# JAPAN PATENT OFFICE

This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: October 30, 2002

Application Number: No. 2002-316806

[JP2002-316806]

Applicant(s): Yamaha Corporation

July 18, 2003

Commissioner,

Japan Patent Office    Yasuo IMAI (Seal)

Certification Number: Pat 2003-3057334

| | |
|---|---|
| [ Document Name ] | Patent Application |
| [ Reference Number ] | C 30767 |
| [ Filing Date ] | 14th year of Heisei, October 30 |
| [ Acceptor ] | To the Director General of the Patent Office |
| [ International Patent Classification ] | G11B 31 / 00 |
| [ Title of the Invention ] | Apparatus for synchronized playback of audio data and performance data |
| [ Number of Claims ] | 6 |
| [ Inventor ] | |
| [ Address ] | c/o YAMAHA CORPORATION, 10-1, Nakazawa-cho, Hamamatsu-shi, Shizuoka-ken |
| [ Name ] | Yoshihiro SHIIYA |
| [ Inventor ] | |
| [ Address ] | c/o YAMAHA CORPORATION, 10-1, Nakazawa-cho, Hamamatsu-shi, Shizuoka-ken |
| [ Name ] | Takeyoshi AIHARA |
| [ Applicant for Patent ] | |
| [ Discriminative Number ] | 000004075 |
| [ Name ] | YAMAHA CORPORATION |
| [ Attorney ] | |
| [ Discriminative Number ] | 100098084 |
| [ Patent Attorney ] | |
| [ Name ] | Kenji KAWASAKI |
| [ Indication of Official Fee ] | |
| [ Ledger Code of Prepayment ] | 038265 |
| [ Amount of Money ] | 21000 |
| [ List of Documents Attached ] | |
| [ Document ] | Specification 1 |
| [ Document ] | Drawings 1 |
| [ Document ] | Abstract 1 |

[ DOCUMENT NAME ] Specification

[ TITLE OF THE INVENTION ] Apparatus for synchronized playback of audio data and performance data and method therefor

[ SCOPE OF THE PATENT CLAIM ]

[ Claim 1 ]

A player being characterized by having

a performance data receiving means for receiving performance data comprising control data for instructing control of performance and performance timing data instructing an execution timing of said control of the performance;

a first adjusted data receiving means for receiving a first adjusted data;

a second adjusted data receiving means for receiving a second adjusted data;

a timing adjusting means for expanding or compressing a time axis of the aforesaid performance timing data by using a ratio between a difference between different two points along the time axis of the aforesaid performance timing data and a difference between a first timing serving as a timing specified by the aforesaid first adjusted data and a second timing serving as a timing specified by the aforesaid second adjusted data, and for adjusting transmission timings of respective sets of data of the aforesaid control data represented by the aforesaid performance timing data by moving forward or backward the aforesaid time axis based on the aforesaid first timing and the aforesaid second timing; and

a control data transmission means for transmitting the respective sets of data of the aforesaid control data based on the adjusted transmission timing by the aforesaid timing adjusting means.

[ Claim 2 ]

The player according to claim 1 being characterized by further having an offset

data receiving means for receiving a first offset data and a second offset data; and

wherein the aforesaid first timing is specified by the aforesaid first offset data in addition to the aforesaid first adjusted data and the aforesaid second timing is specified by the aforesaid second offset data in addition to the aforesaid second adjusted data.

[ Claim 3 ]

The player according to claim 1 being characterized by further having:

an audio data receiving means for receiving audio data representing an audio waveform of a musical tune; and

an audio data playback means for playing back the aforesaid audio data.

[ Claim 4 ]

The player according to claim 1 being characterized by further having a recording means for recording a third offset data representing the aforesaid first timing data and a fourth offset data representing the aforesaid second timing data together with the aforesaid performance data.

[ Claim 5 ]

A playback method being characterized by having:

a performance data receiving step for receiving performance data comprising control data for instructing control of performance and performance timing data instructing an execution timing of said control of the performance;

a first adjusted data receiving step for receiving a first adjusted data;

a second adjusted data receiving step for receiving a second adjusted data;

a timing adjusting step for expanding or compressing a time axis of the aforesaid performance timing data by using a ratio between a difference between different two points along the time axis of the aforesaid performance timing data and a difference between a

first timing serving as a timing specified by the aforesaid first adjusted data and a second timing serving as a timing specified by the aforesaid second adjusted data, and for adjusting transmission timings of respective sets of data of the aforesaid control data represented by the aforesaid performance timing data by moving forward or backward the aforesaid time axis based on the aforesaid first timing and the aforesaid second timing; and

a control data transmission step for transmitting the respective sets of data of the aforesaid control data based on the adjusted transmission timing by the aforesaid timing adjusting step.

[ Claim 6 ]

A program causing a computer to execute

a performance data receiving process for receiving performance data comprising control data for instructing control of performance and performance timing data instructing an execution timing of said control of the performance;

a first adjusted data receiving process for receiving a first adjusted data;

a second adjusted data receiving process for receiving a second adjusted data;

a timing adjusting process for expanding or compressing a time axis of the aforesaid performance timing data by using a ratio between a difference between different two points along the time axis of the aforesaid performance timing data and a difference between a first timing serving as a timing specified by the aforesaid first adjusted data and a second timing serving as a timing specified by the aforesaid second adjusted data, and for adjusting transmission timings of respective sets of data of the aforesaid control data represented by the aforesaid performance timing data by moving forward or backward the aforesaid time axis based on the aforesaid first timing and the aforesaid second timing; and

a control data transmission process for transmitting the respective sets of data of

the aforesaid control data based on the adjusted transmission timing by the aforesaid timing adjusting process.

[ DETAILED EXPLANATION OF THE INVENTION ]

[ 0001 ]

[ TECHNICAL FIELD OF THE INVENTION ]

The present invention relates to an apparatus and a method for playing back a performance data including information relating to performance control of a musical tune in synchronization with playback of an audio data.

[ 0002 ]

[ PRIOR ART ]

There is an apparatus for reading out audio data from a recording medium such as a music CD (Compact Disc) and generating sounds from the readout audio data to be output as a means for playing back a musical tune. There is an automatic performance apparatus for reading out data including information on performance control of a musical tune from a recoding medium such as an FD (Floppy Disk) and for controlling tone generation of a tone generator by using the readout data as another means for playing back the musical tune. There is MIDI data created by complying with the MIDI (Musical Instrument Digital Interface) standard as the data including information relating performance control of the musical tune.

[ 0003 ]

If it is possible to synchronize the automatic performance by the MIDI data with the playback of the audio data recorded in the music CD which is commercially available in general, a preferable accompaniment with the MIDI data will be possible for the played back music CD and it is convenient.

[ 0004 ]

There is a conventional technology, which is applicable to the synchronization between a commercially available CD and the performance by the MIDI data, such that contents of effects such as lighting, image, sounding are previously written in the performance data and the various effects are controlled at correct timings by using flags recorded in the performance data in association with events generated by the performance (see, for example, Patent Publication 1). There is also a conventional technology for synchronization of the MIDI data such that synchronization data are supplemented between the MIDI events asynchronously generated.

[ 0005 ]

[ Patent Publication 1 ]

Patent Publication of Unexamined Application No. 2001 - 195061

[ Patent Publication 2 ]

Patent Application No. 2001 - 215958

[ 0006 ]

[ PROBLEM TO BE SOLVED BY THE INVENTION ]

According to the conventional technologies, the audio data of the music CD is started to be played back at correct timings with respect to the MIDI data. However, a mere start of playback of the audio data of the music CD at a correct timing does not alleviate a problem of the shift between a musical tune played back from the music CD and the performance through the playback of the MIDI data due to the slight difference in a clock speed used during the recording of the MIDI data and a clock speed used during the playback of the MIDI data.

[ 0007 ]

Moreover, when there are different versions of music CDs for a single music tune, a silent period from the playback start of the audio data to the actual start timing of the musical tune may be different from version to version. Further, the clock speed used during the recording may be different from each other for the audio data of the different version of music CD. By virtue of this, the speeds of the music tune are slightly different from each other even when they are played back with the same player. The reason for the difference in the silent period or speed is because the audio data recorded in the music CD are re-recorded after editing the acoustic effects of the master data recording the actual performance. In other words, a part of the silent period is cut during the editing and the clock speed used during the re-recording of the audio data after editing is not precisely similar thereto so that the silent period and the speed are different from version to version.

[ 0008 ]

When there are plural versions of music CDs for the musical tune as mentioned the above, different playback start timings need to be set for different versions. Even when there is no time shift between the recording and the playback of the MIDI data, the musical tune in the audio data is played earlier or slower than the MIDI data depending on the version. Accordingly, different sets of MIDI data need to be prepared for respective versions for the same musical tune and it is inconvenient.

[ 0009 ]

By contemplating the above mentioned circumstances, it is an object of the present invention to provide a player, a playback method and program of the performance data such as MIDI data which is synchronously played back with the plural versions of the audio data for the same musical tune.

[ 0010 ]

[ MEANS TO SOLVE THE PROBLEM ]

To solve the above explained problems, the present invention provides a player being characterized by having:

a performance data receiving means for receiving performance data comprising control data for instructing control of performance and performance timing data instructing an execution timing of said control of the performance;

a first adjusted data receiving means for receiving a first adjusted data;

a second adjusted data receiving means for receiving a second adjusted data;

a timing adjusting means for expanding or compressing a time axis of the aforesaid performance timing data by using a ratio between a difference between different two points along the time axis of the aforesaid performance timing data and a difference between a first timing serving as a timing specified by the aforesaid first adjusted data and a second timing serving as a timing specified by the aforesaid second adjusted data, and for adjusting transmission timings of respective sets of data of the aforesaid control data represented by the aforesaid performance timing data by moving forward or backward the aforesaid time axis based on the aforesaid first timing and the aforesaid second timing; and

a control data transmission means for transmitting the respective sets of data of the aforesaid control data based on the adjusted transmission timing by the aforesaid timing adjusting means.

[ 0011 ]

In a preferred embodiment, the player implemented by the present invention is characterized by further having an offset data receiving means for receiving a first offset data and a second offset data; and

wherein the aforesaid first timing is specified by the aforesaid first offset data in

addition to the aforesaid first adjusted data and the aforesaid second timing is specified by the aforesaid second offset data in addition to the aforesaid second adjusted data. .

[ 0012 ]

In another preferred embodiment, the player implemented by the present invention is characterized by further having:

an audio data receiving means for receiving audio data representing an audio waveform of a musical tune; and

an audio data playback means for playing back the aforesaid audio data.

[ 0013 ]

In still another preferred embodiment, the player implemented by the present invention is characterized by further having a recording means for recording a third offset data representing the aforesaid first timing data and a fourth offset data representing the aforesaid second timing data together with the aforesaid performance data.

[ 0014 ]

The present invention provides a playback method being characterized by having:

a performance data receiving step for receiving performance data comprising control data for instructing control of performance and performance timing data instructing an execution timing of said control of the performance;

a first adjusted data receiving step for receiving a first adjusted data;

a second adjusted data receiving step for receiving a second adjusted data;

a timing adjusting step for expanding or compressing a time axis of the aforesaid performance timing data by using a ratio between a difference between different two points along the time axis of the aforesaid performance timing data and a difference between a first timing serving as a timing specified by the aforesaid first adjusted data and a second

timing serving as a timing specified by the aforesaid second adjusted data, and for adjusting transmission timings of respective sets of data of the aforesaid control data represented by the aforesaid performance timing data by moving forward or backward the aforesaid time axis based on the aforesaid first timing and the aforesaid second timing; and

a control data transmission step for transmitting the respective sets of data of the aforesaid control data based on the adjusted transmission timing by the aforesaid timing adjusting step.

[ 0015 ]

The present invention provides a program causing a computer to execute

a performance data receiving process for receiving performance data comprising control data for instructing control of performance and performance timing data instructing an execution timing of said control of the performance;

a first adjusted data receiving process for receiving a first adjusted data;

a second adjusted data receiving process for receiving a second adjusted data;

a timing adjusting process for expanding or compressing a time axis of the aforesaid performance timing data by using a ratio between a difference between different two points along the time axis of the aforesaid performance timing data and a difference between a first timing serving as a timing specified by the aforesaid first adjusted data and a second timing serving as a timing specified by the aforesaid second adjusted data, and for adjusting transmission timings of respective sets of data of the aforesaid control data represented by the aforesaid performance timing data by moving forward or backward the aforesaid time axis based on the aforesaid first timing and the aforesaid second timing; and

a control data transmission process for transmitting the respective sets of data of the aforesaid control data based on the adjusted transmission timing by the aforesaid

timing adjusting process.

[ 0016 ]

Using such configured apparatus, method and program, the user freely determines the playback timing of the control data and the playback speed upon playing back the audio data. As a result, the synchronized playback of the audio data and the control data is realized.

[ 0017 ]

[ EMBODIMENT OF THE INVENTION ]

[ 1: Embodiment ]

Hereunder, the explanation will be made on an apparatus for realizing the synchronization between the playback of the audio data recorded on a music CD and the playback of the performance by the MIDI data recorded on an FD and the operation of such apparatus. The audio data available for the present invention is not limited to the audio data recorded in the music CD and any type of audio data is utilized. Further, the performance data available for the present invention is not limited to the MIDI data and any type of performance data is available.

[ 0018 ]

[ 1.1: Structure, function and data format ]

[ 1.1.1: Whole configuration ]

Fig. 1 is a view to show a configuration of a synchronized recorder and player SS implemented by an embodiment of the present invention. The synchronized recorder and player SS comprises a CD drive 1, an FD drive 2, an automatic player piano 3, a tone generating portion 4, a manipulating display 5, and a controller 6. The CD drive 1, FD drive 2, automatic player piano 3, tone generating portion 4 and manipulating display 5 are

connected to the controller 6 by communication lines, respectively. The automatic player piano 3 and the tone generating portion 4 are directly connected each other by the communication line.

[ 0019 ]

[ 1.1.2: CD drive ]

The audio data stored in the music CD includes audio data representing audio information and time codes representing playback timings of the audio data and table of contents information such as starting time or the like of respective audio data. The CD drive 1 reads out the audio data from the music CD under instructions from the controller 6 and sequentially outputs the readout audio data. The CD drive 1 is connected to a communication interface 65 in the controller 6 by a communication line.

[ 0020 ]

The audio data, which is output from the CD drive 1, is 16 bit digital audio data in two channels in left and right quantized at a sampling frequency of 44,100 Hz in 16 bits. The data output from the CD drive 1 does not include the time code. Since the configuration of the CD drive 1 is similar to a general CD drive which is capable of outputting the digital audio data, the explanation will be omitted.

[ 0021 ]

[ 1.1.3: FD drive ]

The FD drive 2 records SMF (Standard MIDI File) in the FD or reads out the SMF recorded in the FD in order to transmit the readout SMF. The FD drive 2 is connected to the communication interface 65 in the controller 6 by the communication line. Since the configuration of the FD drive 2 is similar to a general FD drive, the explanation will be omitted.

[ 0022 ]

[ 1.1.4: Event data and SMF ]

The SMF is a file including the event data serving as the performance control content complied with the MIDI standard and delta time serving as data representing the execution timing of the respective event data. The event data and the format of the SMF will be explained with reference to Fig. 2 and Fig. 3.

[ 0023 ]

Fig. 2 shows the general overview of the SMF format. The SMF includes a header chunk and a track chunk. The header chunk includes control data relating data format and time unit included in the track chunk. The track chunk includes the event data and delta time representing an execution timing of respective event data. In the following explanation, the delta time is represented by an absolute time from a base time and is expressed by seconds as a unit, however, the delta time is expressed by the format complied with the MIDI standard. The delta time may be expressed by a relative time and by another expression.

[ 0024 ]

A note-on event, a note-off event and a system exclusive event are shown in Fig. 3 as examples of event data in the SMF. The event data other than the system exclusive event are called as "performance data" in order to discriminate them from the system exclusive event. The event data does not include the time information and is utilized for controlling the tone generation, tone extinction and other control of the musical tones in real time.

In the present specification, the MIDI data is the comprehensive name for data created by complying with the MIDI standard in the present specification.

[ 0025 ]

[ 1.1.5: Automatic player piano ]

The automatic player piano 3 is a musical tone generator which outputs an acoustic piano tone and an electronically synthesized piano tone in response to a key manipulation and a pedal manipulation by the user of the synchronized recorder and player SS. The automatic player piano 3 generates a performance event in response to the key manipulation and the pedal manipulation by the user and transmits the generated performance event. Further, the automatic player piano 3 receives the MIDI event and automatically plays with acoustic piano sounds and electronically synthesized piano tones in response to the received performance events.

[ 0026 ]

The automatic player piano 3 comprises a piano 31, key sensors 32, pedal sensors 33, a MIDI event control circuit 34, a tone generator 35 and a driving part 36.

[ 0027 ]

The key sensor 32 and the pedal sensor 33 are provided on each of the plural keys and the plural pedals of the piano 31 in order to detect the positions of keys and pedals, respectively. The key sensor 32 and pedal sensor 33 transmit the detected position information, identification number corresponding to each of the keys and pedals, respectively, and detected time information to the MIDI event control circuit 34.

[ 0028 ]

The MIDI event control circuit 34 receives the position information of the keys and pedals, respectively, from the key sensors 32 and the pedal sensors 33 together with the identification information of the keys and pedals and the time information, and immediately generates performance events such as a note-on event, note-off event or the

like from the information in order to output the generated performance event to the controller 6 and the tone generator 35. The MIDI event control circuit 34 receives the performance event from the controller 6 and transfers the received performance event to the tone generator 35 or the driving part 36. Further, the MIDI event control circuit 34 is under the instruction of the controller 6 to determine which the performance event received from the controller 6 is transferred to the tone generator 35 or the driving part 36.

[ 0029 ]

The tone generator 35 receives the performance events from the MIDI event control circuit 34 and outputs the sound information of various musical instruments as digital audio data in the left and right channels based on the received performance events. The tone generator 35 electronically synthesizes the digital audio data at a pitch designated by the received performance event and transmits it to a mixer 41 in the tone generating portion 4.

[ 0030 ]

The driving parts 36 are provided on the respective keys and pedals of the piano 31 and comprises a group of solenoids for driving these and a control circuit for controlling the group of solenoids. When the control circuit of the driving parts 36 receives the performance events from the MIDI event control circuit, it adjusts current to be supplied to the solenoid provided on a corresponding key or pedal in order to adjust magnetic flux generated by the solenoid and the key or pedal is operated in response to the performance event.

[ 0031 ]

[ 1.1.6: Tone generator ]

The tone generating portion 4 receives the audio data from the automatic player

piano 3 and the controller 6 and converts the received audio data into sounds to be output. The tone generating portion 4 comprises the mixer 41, a D/A converter 42, an amplifier 43 and a speaker 44.

[ 0032 ]

The mixer 41 is a digital stereo mixer which receives plural sets of digital audio data in the two channels, left and right, and converts these into a pair of left and right digital audio data. The mixer 41 receives the digital audio data from the tone generator 35 of the automatic player piano 3 and at the same time, receives the digital audio data, which is read out by the CD drive 1 from the music CD, through the controller 6. The mixer 41 calculates an average of the received digital audio data and transmits this to the D/A converter 42 as a pair of digital audio data in right and left.

[ 0033 ]

The D/A converter 42 receives the digital audio data from the mixer 41 and converts the received digital audio data into the analog audio signal to be output to the amplifier 43. The amplifier 43 amplifies the analog audio signal, which is input from the D/A converter 42, and outputs it to the speaker 44. The speaker 44 converts the analog audio signal, which is input from and amplified by the amplifier 43, into the sounds. As a result, the audio data recorded in the music CD and the audio data generated by the tone generator 35 are output from the tone generating portion 4 as stereo sounds.

[ 0034 ]

[ 1.1.7: Manipulation display ]

The manipulation display 5 is a user interface when a user of the synchronized recorder and player SS manipulates the synchronized recorder and player SS. The manipulation display 5 includes key pads when the user depresses to give instructions to

the synchronized recorder and player SS and a liquid crystal display for confirming the state of the synchronized recorder and player SS. When the key pad is depressed by the user, the manipulation display 5 outputs a signal corresponding to the depressed key pad to the controller 6. The manipulation display 5 receives bit map data including information of characters and figures and displays the characters and figures based on the received bit map data on the liquid crystal display.

[ 0035 ]

[ 1.1.8: Controller ]

The controller 6 controls the entire synchronized recorder and player SS and the configuration is similar to a general purpose computer. The controller 6 comprises a ROM (Read Only Memory) 61, a CPU (Central Processing Unit) 62, a DSP (Digital Signal Processor) 63, a RAM (Random Access Memory) 64 and a communication interface 65. The components are mutually connected each other through a bus. The controller 6 has a clock which is not shown in the drawing and the operations of the respective components are synchronized by clock signals generated by the clock.

[ 0036 ]

The ROM 61 is a non-volatile memory for storing various kinds of control program. The control program, which is stored in the ROM 61, includes program for general control routine and program which causes the CPU 62 to execute routines for recording operations and playback operations of the SMF which will be mentioned later. The CPU 62 is a microprocessor, which executes general purpose processings, and reads out the control program from the ROM 61 and executes the control routines in accordance with the readout control program. The DSP 63 is a microprocessor, which processes the digital audio data at a high speed, executes data generation routine for correlation

discrimination and filter operation necessary for the correlation discrimination routine under the control of the CPU 2, which will be mentioned later, on the digital audio data received from the CD drive 1 and the FD drive 2 by the controller 6, and transmits the resulting data to the CPU 62. The RAM 64 is a volatile memory and temporarily stores the data used by the CPU 62 and DSP 63. The communication interface 65 is an interface which is capable of transmitting and receiving the digital data in various formats, converts the format necessary for digital data transmitted or received between the CD drive 1, FD drive 2, automatic player piano 3, tone generating portion 4, and manipulation display 5, and relays the data between the respective devices and the controller 6.

[ 0037 ]

[ 1.2: Operation ]

Next, operations of the synchronized recórder and player SS will be explained.

[ 1.2.1: Recording operation ]

First, operations of the synchronized recorder and player SS, when a user of the synchronized recorder and player SS plays a piano in synchronization with the playback of a commercially available music CD and the information of the performance is recorded on an FD as the SMF, will be explained. The music CD used during the recording operation, which will be explained below, is called as a music CD-A in order to discriminate the music CD used during the playback operation mentioned later. The audio data of the plural musical tune are stored in the music CD-A and it is assumed that the recording operation will be done for an audio data NA of a certain musical tune N among those in the following explanation.

[ 0038 ]

[ 1.2.1.1: Start operation of recording ]

The user sets the music CD-A in the CD drive 1 and an FD having a sufficient capacity in the FD drive 2. Subsequently, the user depresses the key pads of the manipulation display 5 and instructs the recording start of the performance data corresponding to the musical tune N included in the music CD-A. The manipulation display 5 outputs the signal corresponding to the depressed key pad to the controller 6.

[ 0039 ]

The CPU 62 of the controller 6 receives the signal corresponding to the recording start of the performance data from the manipulation display 5 and transmits a playback instruction of the audio data NA to the CD drive 1. In response to the playback instruction, the CD drive 1 reads out the audio data NA from the music CD-A and sequentially transmits the readout audio data NA to the controller 6.

[ 0040 ]

The controller 6 receives the data for a pair of right and left channels for every 1/44100 second from the CD drive 1. Hereunder, the data values for a pair of the right and left channels are expressed as (R(n), L(n)), and the pair of data values or respective sets of data values generated from the pair of data values in the data generation process for correlation discrimination or a management event generation process are called as "sample values". It is expressed as a sample value (n) in order to discriminate different sample values. n is an integer representing an order of the audio data and increases from the start of the data such as 0, 1, 2 .... The R(n) and L(n) represent data values in the right channel and the left channel, respectively, and they are either of integers ranging from - 32768 to 32767.

[ 0041 ]

[ 1.2.1.2: Transmission of audio data to the tone generating portion ]

After receiving a first sample value, namely, a sample value (0), of the audio data NA from the CD drive 1, the CPU 62 starts measuring time from the timing (hereunder referred to as "base timing P") in accordance with clock signals acquired from the clock. In other words, it is 0.00 second at the base timing P.

[ 0042 ]

The CPU 62 sequentially receives a sample value (1), a sample value (2), .... The CPU 62 sequentially transmits the sample values to the tone generating portion 4. The tone generation portion 4 receives the sample values and converts them into sounds for output. As a result, the user can listen to the musical tune N. The CPU 62 continues the transmission process until the last sample value of the musical tune N is transmitted to the tone generating portion 4.

[ 0043 ]

[ 1.2.1.3: Storing audio data in queue ]

The CPU 62 sequentially transmits the received sample values to the tone generating portion 4 and at the same time, stores the received sample values in the queue in the RAM 64 together with time information representing the received time of the sample values. The time information is a lapse of time from the base timing P. Hereunder, the time information corresponding to the sample value (n) is called as time information (n).

[ 0044 ]

In the present embodiment, the 1,323,000 pairs of sample values are available to be stored in the queue and the CPU 62 receives a new sample value after the number of sample values reaches 1,323,000 pairs which have been already stored in the queue and then deletes the sample value at the start of the queue and stores the newly received sample value at the end of the queue. The 1,323,000 pairs of sample values stored in the queue

amount to audio data for 30 seconds. The CPU 62 continues the process of storing the sample values and time information in the queue until the CPU 62 receives the last sample value of the audio data NA.

[ 0045 ]

[ 1.2.1.4: Storing raw audio data for start reference ]

The CPU 62 continues the transmission and the storage of the above mentioned sample values to the tone generating portion 4 and into the queue, respectively, and separately stores sample values corresponding to a certain period of time after an actual starting timing of a musical tune in the audio data into the RAM 64. In the present embodiment, it is assumed that the sample values stored in this process is $2^{16}$ pairs, namely, 65536 pairs, as an example. The 65536 pairs of sample values correspond to the audio data for about 1.49 seconds. Hereunder, the 65536 pairs of sample values are called as "raw audio data for start reference". Next, a storage process of the raw audio data for start reference will be explained.

[ 0046 ]

First, the CPU 62 sequentially judges whether an absolute value of either of the left and right sample values exceeds a predetermined threshold value or not with respect to each of the received sample values. Hereunder, the judgment process is called as "threshold judgment process". The threshold is assumed to be 1000 in the present embodiment, for example. Accordingly, if the absolute value of either of R(n) and L(n) is larger than 1000, the CPU 62 acquires an affirmative result in the threshold judgment process.

[ 0047 ]

For example, it is assumed that the absolute value of R(50760) or L(50760),

which corresponds to the sample value (50760) of the audio data NA, exceeds 1000 for the first time. In this case, the CPU 62 acquires negative results in the threshold judgment process for the sample value (0) to sample value (50759). This shows that there is a silent or a substantially silent portion at the beginning of the audio data NA for about 1.15 seconds.

[ 0048 ]

The CPU 62 acquires the positive result for the received sample value in the threshold judgment process and then, skips the threshold judgment process for the sample values received thereafter and stores the time information representing the received timing of the sample value having the base timing P as the starting point. The time information is called as "start time information" hereinafter. Further, the CPU 62 stores 65536 pairs of sample values received after the sample value, which the threshold judgment process results in positive, in the RAM 64 as the raw audio data for start reference.

[ 0049 ]

For example, when the threshold judgment process gives a positive result for the sample value (50760) for the first time, the raw audio data for start reference are a sample value (50760) to a sample value (116295) and the start time information is the time information representing the receiving time of the sample value (50760), namely, for about 1.15 seconds.

[ 0050 ]

[ 1.2.1.5: Generation of processed audio data for start reference ]

The CPU 62 finishes storing the raw audio data for start reference into the RAM 64 and transmits an execution instruction of the data generation process for correlation discrimination on the raw audio data for start reference. The data generation process for

correlation discrimination is a process for generating audio data sampled at a sampling frequency of about 172.27Hz for correlation discrimination process from the audio data sampled at a sampling frequency of 44,100Hz. The correlation discrimination process is a process to judge similarity of two pairs of audio data and the details will be mentioned later. Hereunder, the data generation process for correlation discrimination will be explained with reference to Fig. 4.

[ 0051 ]

The DSP 63 receives the instruction to execute the data generation process for correlation discrimination on the raw audio data for start reference from the CPU 62 and reads out the reference raw audio data for start reference from the RAM 64 (step S1). Subsequently, the DSP 63 calculates an arithmetic average of the left and right values to the respective sample values of the raw audio data for start reference and converts the stereo data into a monaural data (step S2). The conversion process into the monaural is a process to reduce the workload on the DSP 63 in processes after this step.

[ 0052 ]

Subsequently, the DSP 63 puts a series of sample values converted into the monaural signal in a high pass filtering (step S3). The DC components in the audio waveform represented by the series of sample values are eliminated by this high pass filtering and the sample values are uniformly distributed in positive and negative sides. Two pairs of audio data are compared and discriminated based on cross correlation values in the correlation discrimination process, and the preciseness of discrimination is enhanced if the sample values are uniformly distributed on positive and negative sides when the cross correlation values are compared. In other words, the process in this step is a process to improve the accuracy of the judgment in the correlation discrimination process.

[ 0053 ]

Subsequently, the DSP 63 calculates absolute values of respective sample values after the high pass filtering (step S4). The process in the step calculates substitute values of power of the respective samples. Since the absolute values have smaller values than square values representing the power and are easily processed, the present embodiment uses the absolute values instead of the square values of respective sample values. The square values may be calculated instead of the absolute values of the respective sample values in this step when the performance of the DSP 63 is high.

[ 0054 ]

Subsequently, the DSP 63 filters the series of sample values, which are converted into the absolute values in the step S4, through a comb filter (step S5). The process in this step extracts the low frequency components, of which the variation in the waveform is easily to be detected, from the audio signal waveform represented by the series of sample values. A low pass filter is normally used to extract the low frequency components; since the comb filter applies less load to the DSP 63 than the low pass filter, the comb filter is replaced with the low pass filter in the present embodiment.

[ 0055 ]

Fig. 5 shows a configuration of an example of the comb filter to be employed in the step S5. In Fig. 5, a process represented by a square rectangular means a delay process and k in $z^{-k}$ means that the delay time in the delay process is (sampling cycle X k). As mentioned previously, the sampling frequency of the music CD is 44100 Hz and the sampling period is 1 / 44100 second. On the other hand, the process represented by a triangle means a multiplication and a value indicated in the triangle means a coefficient of the multiplication. In Fig. 5, K is expressed by a following expression (1).

[ Expression 1 ]

$$K = \frac{44100 - \pi \times f}{44100 + \pi \times f} \quad \text{.........} (1)$$

[ 0056 ]

The multiplication using K as a coefficient gives the comb filter a function of a high pass filter having a cutoff frequency f. As a result, the DC components in the audio waveform represented by the series of sample values are eliminated again by the filtering process in this step. Moreover, the values of k and f are arbitrarily varied and are empirically calculated in order to enhance the accuracy of discrimination in the correlation discrimination process.

[ 0057 ]

Subsequently, the DSP 63 filters the series of sample values, which are filtered in the step S5 in Fig. 4, through a low pass filter (step S6). The process in this step avoids aliasing noise in a down sampling process rendered in a next step S7. Since the data at the sampling frequency of 44100 Hz are sampled down to a sampling frequency of about 172.27 Hz, the frequency components of about 86.13 Hz, which is a half thereof, or higher need to be eliminated in order to avoid the aliasing noise. However, the high frequency components are not sufficiently eliminated in the filtering process in the step S5 using the comb filter due to the characteristics of the comb filter. Accordingly, the remaining frequency components of about 86.13 Hz or higher are eliminated by the filtering process using the low pass filter in this step. If the performance of the DSP 63 is high, a filtering process using a single low pass filter with a high accuracy is acceptable instead of the filtering process using two filters in the step S5 and step S6.

[ 0058 ]

Subsequently, the DSP 63 samples down the series of sample values filtered in the

step S6 by 1/256 (step S7). In other words, the DSP 63 extracts one sample value from every 256 sample values. As a result, the number of the series of sample data is reduced from 65536 to 256. Hereunder, a series of 256 sample values acquired from the process in the step S7 is called as "processed audio data for start reference". The DSP 63 stores the processed audio data for start reference in the RAM 64 (step S8). The process in the step S7 is a process to reduce the load on the DSP 63 by the process utilizing the processed audio data for start reference mentioned later and to reduce the recording size of the processed audio data for start reference into the FD. Accordingly, when the processing performance of the DSP 63 is high or the available storage capacity is sufficient, the process of the step S7 is not executed and the sample value representing the audio data at 44,100 Hz itself may serve as the processed audio data for start reference.

[ 0059 ]

[ 1.2.1.3 ] Generation of management event

The CPU 62 receives the start sample value of the raw audio data for start reference and stores the above mentioned raw audio data for start reference in the RAM 64 and, at the same time, transmits an execution instruction for a management event generating process, to the DSP 63. The management event generation process is a process including a filter process extracting frequency components equal to or lower than a certain frequency on the audio waveform represented by the series of sample values stored in the queue of the RAM 64 and a filter process extracting frequency components equal to or lower than a frequency which is lower than the frequency, and compares the values acquired by the two filter processes in order to generate the management event. The management event is a flag to generate time information used in a timing adjustment process of the performance data mentioned later. The management event generating

process will be explained with reference to Fig. 6.

[ 0060 ]

The DSP 63 receives an execution instruction of the management event generation process from the CPU 62 and reads out a certain number of samples from the last sample value stored in the queue of the RAM 64 (step S11). In the present embodiment, it is assumed that the DSP 63 reads out 44,100 pairs of sample values from the queue. Hereunder, the series of sample values, which are read out from the queue in the management event generation process by the DSP 63, are called as "raw audio data" and the raw audio data with the sample value (n) as the last one is represented as "raw audio data (n)" if the plural sets of raw audio data have to be discriminated.

[ 0061 ]

For example, when the last sample value stored in the queue when the DSP 63 receives the execution instruction of the management event generation process is a sample value (50760), a first raw audio data is a raw audio data (50760), namely, a sample value (6601) to a sample value (50760).

[ 0062 ]

Subsequently, the DSP 63 executes an arithmetic average operation of the left and right values of the respective sample values of the raw audio data and converts the stereo data into a monaural data (step S12). The conversion process into the monaural is a process to reduce the workload on the DSP 63 in processes after this step.

[ 0063 ]

Subsequently, the DSP 63 calculates absolute values for the series of sample values converted into the monaural signal (step S13). The process in the step calculates substitute values of power of the respective samples. Since the absolute values have

smaller values than square values representing the power and are easily processed, the present embodiment uses the absolute values instead of the square values of respective sample values. Accordingly, the square values may be calculated instead of the absolute values of the respective sample values in this step when the performance of the DSP 63 is high.

[ 0064 ]

Subsequently, the DSP 63 filters the series of sample values converted into the absolute values in the step S13 through the low pass filter (step S14). Hereinafter, the values acquired as a result of the processes in the steps S11 to S14 are called as "middle term index" and the middle term index corresponding to the sample value (n) is called as a "middle term index (n)". Herein, a cut-off frequency of the low pass filter used in the step S14 is assumed to be at 100 Hz. The middle term index (n) indicates a tendency in a middle term in a variation of the audio waveform represented by the series of sample values at the timing corresponding to the sample value (n). In other words, the audio waveform represented by the series of sample values goes up and down in a short term and the variations in values of the series of sample values filtered by the low pass filter are suppressed by the plural precedent sample values. As a result, the short term variation component is removed from the waveform represented by the series of the middle term index and the variation in the middle term component and the long term component remains. The series of the middle term indices including the middle term index (n) as the last one, namely, ... the middle term index (n-2), the middle term index (n-1) and the middle term index (n) are created by the process in the step S4. The DSP 63 stores these middle term indices in the RAM 64 (step S15).

[ 0065 ]

Subsequently, the DSP 63 filters the series of the middle term indices acquired in the step S14 through a comb filter (step S16). The process in this step is a process which extracts the low frequency component equal to or lower than a certain frequency from the audio waveform represented by the series of the middle term indices. This step is similar to the process using the low pass filter having the cut-off frequency lower than the cut-off frequency of the low pass filter used in the step S14, and the comb filter is used instead of the low pass filter in the present embodiment because the comb filter applies smaller load on the DSP 63 than the low pass filter.

[ 0066 ]

Fig. 7 shows a configuration of an example of the comb filter to be employed in the step S16. In Fig. 7, a process represented by a square rectangular means a delay process and k in $z^{-k}$ means that the delay time in the delay process is (sampling cycle X k). As mentioned previously, the sampling frequency of the music CD is 44100 Hz and the sampling period is 1 / 44100 seconds. On the other hand, the process represented by a triangle means a multiplication and a value indicated in the triangle means a coefficient of the multiplication. Hereinafter, it is assumed that k = 22050, and the frequency components higher than 1 Hz are almost removed therefrom by the filtering process in the step S16. In other words, the audio waveform represented by the series of values acquired by the process in the step S16 is acquired by extracting the long term variation component by removing the middle term variation component from audio waveform represented by the series of the middle term indices.

[ 0067 ]

Subsequently, the DSP 63 multiplies the series of values acquired by the filtering process in the step S16 in Fig. 6 by a positive constant h, respectively (step S17). The

multiplication process by the constant h is a process to adjust the frequency to acquire the positive result in a comparing process in a next step S18 and time intervals which acquires the positive results in the comparing process generally become shorter when a value of the constant h becomes smaller. If the time interval becomes too long, time intervals of the generated management events in step S21 below becomes longer and the accuracy of the timing adjusting process of the performance event mentioned later become lower. On the other hand, if the time intervals which the positive results are acquired in the comparing process become too short, the positive results are cancelled by a process in step S20 below one after another; the time intervals of the generated management events also become longer; and as a result, the accuracy of the timing adjusting process of the performance event becomes lower. Accordingly, the value to generate the management events at a proper frequency is empirically used as the value of the constant h.

[ 0068 ]

The value acquired as a result of the process in the step S17 is called as a "long term index" and the long term index corresponding to the sample value (n) are called as a "long term index(n)". A series of the long term indices (n) having the long term index (n) as the last one, namely, ... a long term index (n-2), a long term index (n-1), and a long term index (n) are generated by the process of the step S17. The DSP 63 stores these long term indices in the RAM 64 (step S18).

[ 0069 ]

Subsequently, the DSP 63 reads out the middle term index (n) and the long term index (n) from the RAM 64 and executes the comparison to judge whether the middle term index (n) is equal to or larger than the long term index (n) or not (step S19). This comparison process represents that the audio waveform represented by the raw audio data

varies with a large amplitude in the middle term at a timing corresponding to the sample value (n). In other words, when the volume of the sound included within a frequency band of 1 Hz to 100 Hz in the sound waveform of the musical tune increases rapidly, the value of the middle term index exceeds the value of the long term index so that the positive result (hereinafter referred to as "Yes") is acquired in the comparison process in the step S19.

[ 0070 ]

If the positive result Yes is acquired in the comparison process in the step S19, the DSP 63 stores the time information on a timing measured based on a base timing P in the RAM 64. Subsequently, the DSP 63 reads out the time information when Yes was required in the comparison process in the step S19 in the past from the RAM 64 and judges whether or not there is a record that Yes was acquired in a time period $\tau$ in the past (step S20). When the shorter time intervals cause the result of the comparing process in the step S19 to be Yes, the judgment process in this step S20 avoids generation of management events at shorter time intervals in a next step S21 as similarly. If the management events are generated at shorter time intervals, it is difficult to correctly correlate the recorded management events and management events generated by newly acquired audio data in a timing adjusting process of the performance event mentioned later. The generation of the management events is avoided at time intervals equal to or shorter than the time period $\tau$ by the process in the step S20. The value of $\tau$ is empirically determined in order to set the proper time intervals of generated management events. Further, the first judgment result in the step S20 at an initial judgment becomes No because there is no preceding step S19.

[ 0071 ]

When the negative result (hereinafter referred to as "No") is acquired in the judgment process in the step S20, the DSP 63 transmits a management event, which represents that the audio waveform indicating the raw audio data suffices a predetermined condition at a timing corresponding to the sample value (n) after the above described series of processes, to the CPU 62 (step S21).

[ 0072 ]

If the comparison result in the step S19 becomes No; the judgment result in the step S20 becomes Yes; and the process in the step S21 is finished, the DSP 63 stands by until the CPU 62 receives new sample values, namely, the sample value (n+1), from the CD drive 1 and stores the sample values in the queue of the RAM 64. When the sample value (n+1) are stored in the queue (step S22), the DSP 63 executes processes of the step S11 or after on the raw audio data (n+1) which the sample value (n+1) serves as the last one.

[ 0073 ]

The sequence in the above step S11 to step S22 is continued until the process on the raw audio data, which the last sample value of the audio data NA serves as the last one, finishes. The above is the explanation on the management event generating process.

[ 0074 ]

Fig. 8 is a view to show the management event generation when the management event generation process is executed on the real audio data. Moreover, it is assumed that a single stage IIR (Infinite Impulse Response) filter is used as the low pass filter in the step S14 in order to create the graph. It is assumed that the constant h in the step S17 is 4, and the time period $\tau$ in the step S20 is 0.55 second.

[ 0075 ]

Though there are timings B and C, which the middle term index exceeds the long term index immediately after a timing A when a management event is created, in Fig. 8, the management event is created at the timing A only. Since the timings B does not lapse a predetermined time period, namely, 0.55 second after the timing A and the timing C does not lapse the predetermined time period, the judgment result in step S20 becomes Yes so that the process in the step S21 is not initiated.

[ 0076 ]

[ 1.2.1.7: Generation of performance event ]

The above mentioned management event generation process is executed by the DSP 63 and the user starts the performance on the piano 31. In other words, the user depresses keys and manipulates pedals by accompanying a musical tune N which is output from the tone generating portion 4. The performance information on the piano 31 by the user is detected as the motions of the keys and the pedals through the key sensors 32 and pedal sensors 33, respectively, and is converted into performance events by the MIDI event control circuit 34 to be transmitted to the controller 6.

[ 0077 ]

[ 1.2.1.8: Recording event data ]

As mentioned in the above, the CPU 62 receives the management events and the performance events from the DSP 63 and the MIDI event control circuit 34 of the automatic player piano 3, respectively, during the playback of the audio data NA.

[ 0078 ]

Fig. 9 is a view to show the relationship between the generation of the management event and the generation of the performance event with respect to time. In this case, the CPU 62 receives the management events at the timings lapsing 1.51 seconds,

2.38 seconds, 4.04 seconds, ... from a base point P. The CPU 62 also receives the performance events at the timings lapsing 2.11 seconds, 2.62 seconds, 3.60 seconds, ... from the base point P. For example, the middle term index exceeds the long term index at 1.78 seconds, however, the management event is not received at 1.78 seconds since 0.55 second has not lapsed after the middle term index exceeds the long term index at the timing of 1.78 seconds.

[ 0079 ]

The CPU 62 receives the management event, generates a system exclusive event representing the management event and attaches time information representing a receiving time of the management event with respect to the base point P as a delta time to be stored in the RAM 64. As similarly, the CPU 62 receives the performance event and attaches time information representing a receiving time of the performance event with respect to the base point P as a delta time to be stored in the RAM 64.

[ 0080 ]

[ 1.2.1.9: Specifying end of musical tune ]

The CD drive 1 transmits the last sample value of the audio data NA to the controller 6 and stops the playback of the music CD-A. The CPU 62 receives the last sample value of the audio data NA, finishes the management event generation process on the raw audio data which the sample value serves as the last one, and then, subsequently, executes a process to specify the actual end of the musical tune N in the audio data NA.

[ 0081 ]

When the CPU 62 finishes the management event generation process on the raw audio data which the last sample value of the audio data NA serves as the end, 1323000 pairs of sample values, which the last sample value of the audio data NA serves as the end,

are stored in a queue together with time information representing the receiving time of respective sample values. For example, when it is assumed that the last sample value of the audio data NA is a sample value (7673399), a sample value (6350400) to a sample value (7673399) are stored in the queue together with the time information corresponding to the respective sample values.

[ 0082 ]

First, the CPU 62 reads out the last sample value which is stored in the queue and judges whether an absolute value of either of left or right of the readout sample value exceeds a threshold value, 1000, or not, namely, executes a threshold judgment process. If a negative result is acquired in the threshold judgment, the CPU 62 reads out the sample value which is stored in a penultimate in the queue and executes the threshold judgment process on the readout sample value. The CPU 62 sequentially reads out the sample values from the last to start of the queue until a positive result is acquired in the threshold judgment process and repeats the similar process.

[ 0083 ]

For example, if the an absolute value of R (7634297) or L (7634297) of the sample value (7634297) exceeds 1000 for the first time in the audio data NA having a last sample (7673399), the threshold judgments by the CPU 62 on the sample value (7634298) to the sample value (7673399) go negative. This means that there is a silent or a substantially silent part for about 0.89 second in the last part of the audio data NA.

[ 0084 ]

A sample value giving a positive result in the above mentioned threshold judgment process is hereinafter defined as a sample value (Z). The sample value (Z) is a sample value corresponding to the actual end of the music tune N. When the CPU 62

specifies the sample value (Z) by acquiring the positive result in the threshold judgment process, the threshold judgment process thereafter is not executed.

[ 0085 ]

[ 1.2.1.10: Generation of processed audio data for end reference ]

Subsequently, the CPU 62 executes a generation process of processed audio data for end reference explained hereinafter. The generation process of processed audio data for end reference is defined as a process for generating an audio data for executing a correlation discrimination process mentioned later by the above mentioned correlation identification data generation process on the audio data recorded in the queue.

[ 0086 ]

The data generation process of processed audio data for end reference will be explained with reference to Fig. 10. In the following explanation, a starting of the sample values stored in the queue is defined as a sample value (W). To make the explanation more concretely, it is assumed that W = 6350400 and Z = 7634297. This means that a sample value (6350400) to a sample value (763399) are stored in the queue and the sample value corresponding to the actual end of the musical tune N is the sample value (7634297). The series of 65536 pairs of the sample values having the last sample of sample value (n) is called as "reference raw audio data (n)".

[ 0087 ]

First, the CPU 62 creates a counter i and a counter j and sets i = Z = 7634297 and j = 0 (step S31). Subsequently, the CPU 62 transmits an execution instruction for the data generation process for correlation identification data on a reference raw audio data (i-j) to the DSP 63. Since the data generation process for correlation discrimination is the process which has been already explained with reference to Fig. 4, the explanation is

omitted. The DSP 63 stores a series of 256 sample values acquired by processing the reference raw audio data (i-j) in the RAM 64 in response to the execution instruction of the data generation process for correlation discrimination (step S32). Hereinafter, the 256 sample values is called as "reference processed audio data" and the reference processed audio data acquired from the reference raw audio data (n) are called as "reference processed audio data (n)". Since i-j = 7634297 upon execution of the step S32 for the first time, the reference processed audio data (7634297) is stored in the RAM 64.

[ 0088 ]

Subsequently, the CPU 62 judges whether j = 881999 or not (step S33). The judgment process is a process for reading out a series of 65536 pairs of sample values of the audio data for the last 20 seconds of the audio data NA by shifting the end sample by sample in order to sequentially read out 882000 pairs of reference raw audio data.

[ 0089 ]

Since j = 0, the judgment result in the step S33 becomes No. In this case, the CPU 62 increments the value of j by 1 (step S34) and returns to the process of the step S32. The series of processes of the step S32, step S33 and step S34 are repeated 881999 times and the process of the step S32 is repeated 882000 times. As a result, the RAM 64 stores the reference processed audio data (7634297), reference processed audio data (7634296), ..., reference processed audio data (6752298).

[ 0090 ]

Since j = 881999 in 882000th time of step S33, the judgment result becomes Yes. Subsequently, the CPU 62 transmits an execution instruction of the correlation discrimination process for the reference processed audio data (i-j) of the reference processed audio data (i) stored in the RAM 64 to the DSP 63 (step S35). The correlation

discrimination process is a process to judge the similarity of two pairs of audio data. The correlation discrimination process will be explained with reference to Fig. 11.

[ 0091 ]

The CPU 62 transmits the execution instruction of the correlation discrimination process to the DSP 63 and specifies an original reference audio data for the similarity judgment (hereinafter referred to as "original reference audio data") and a subject reference audio data (hereinafter referred to as "subject reference audio data"). In this case, the original reference audio data is the reference processed audio data (i) and the subject reference audio data is the reference processed audio data (i-j). The 256 sets of data included in the original reference audio data are represented as X(0) to X(255). On the other hand, the 256 sets of data included in the subject reference audio data are represented as Ym(0) to Ym(255). However, m is i-j and represents a number of sample values corresponding to the last reference raw audio data used for generation original audio data.

[ 0092 ]

The DSP 63 receives an execution instruction of the correlation discrimination process from the CPU 62 and reads out the reference processed audio data (i) and the reference processed audio data (i-j) as the original reference audio data and the subject reference audio data, respectively, from the RAM 64 and executes the judgment process expressed by following formula (2) and formula (3) (step S51). It is assumed that i = 7634297, m = i – j = 6752298 in the process of the step S51 executed first.

[ Expression 2 ]

$$\frac{\sum_{i=0}^{255} (X(i) \times Ym(i))}{\sum^{255} (X(i)^2)} \geqq p \ldots\ldots (2)$$

i=0

[ Expression 3 ]

$$\frac{\left\{ \sum_{i=0}^{255} (X(i) \times Ym(i)) \right\}^2}{\sum_{i=0}^{255} (X(i)^2) \times \sum_{i=0}^{255} (Ym(i)^2)} \geqq q \text{ ...... (3)}$$

[ 0093 ]

Making a pair of data having the identical numbers when X(0) to X(255) and Ym(0) to Ym(255) are placed in order, the more the data values coincide therewith the larger the left side of the expression (2) becomes and it approaches to 1. In the following explanation, the value of the left side of the expression is called as an absolute correlation index. The value of p is arbitrarily modified within a range of 0 to 1; it is empirically determined in order to acquire a result "Yes" in the discrimination by the above described expression (2) by using two pairs of data acquired by the data generation process for correlation discrimination on the audio data corresponding to the same part of the same musical tune of the different versions and in order to acquire "No" in the discrimination by the expression (2) by using two pairs of data acquired from the audio data corresponding to the different parts of the musical tune even though they are similar thereto.

[ 0094 ]

The value of the left side of the expression (3) ranges from 0 to 1 and approaches to 1 as shapes of the audio waveshape represented by X(0) to X(255) and the waveshape of the audio waveform represented by Ym(0) to Ym(255) become more similar. The value of the left side of the expression is called as a “relative correlation index” in the following

explanation. The value of the above mentioned absolute correlation index varies, if levels of the audio waveforms represented by X(0) to X(255) and Ym(0) to Ym(255) are different from each other even though the shapes of the audio waveforms represented by X(0) to X(255) and Ym(0) to Ym(255). On the other hand, since the relative correlation index is not influenced by the levels of the audio waveform represented by X(0) to X(255) and Ym(0) to Ym(255) and approximates 1 if the shapes are similar each other, the judgment by the expression (3) gives Yes even if the recording levels are different depending on different versions of the music CDs. The value of q is arbitrarily modified in a range of 0 to 1 and is empirically determined as p.

[ 0095 ]

If both of the results of the two discrimination processes in the step S51 are Yes, the DSP 63 executes discrimination processes expressed by following expression (4) and expression (5) (step S52).

[ Expression 4 ]

$$\frac{d\sum_{i=0}^{255}(X(i) \times Ym(i))}{dm} = 0 \;\ldots\ldots (4)$$

[ Expression 5 ]

$$\frac{d^2\sum_{i=0}^{255}(X(i) \times Ym(i))}{d^2m} < 0 \;\ldots\ldots (5)$$

[ 0096 ]

The left side of the expression (4) is a variation rate of sum of products of X(0) to X(255) with respect to the sample value (n). In the following explanation, the sum of products of X(0) to X(255) and Ym(0) and Ym(255) is called as a “correlation value”.

When the reference processed audio data and the processed audio data for discrimination are arranged in order and data having the same order are paired therewith, the more the pair data values become approximate, the larger the correlation value becomes. The variation ratio of the correlation value becomes 0 when the correlation value becomes an extremum after the correlation values with respect to X(0) to X(255) and Ym(0) to Ym(255) are arranged along the time axis with respect to m. Accordingly, the discrimination process by the expression (4) is a process to judge whether the correlation value is an extremum or not. The process by the expression (5) is to judge whether the extremum is a relative maximum value or not.

[ 0097 ]

Explaining more precisely, since X(0) to X(255) and Ym(0) to Ym(255) are discrete values in the present embodiment, the left side of the expression (4) barely becomes 0. Accordingly, the judgment process in the step S52 is executed as follows. The DSP 63 makes a difference between a sum of products of X(0) to X(255) and Ym(0) to Ym(255) and a sum of products of X(0) to X(255) and Ym-1(0) to Ym-1(255). The value hereunder is called as Dm. Subsequently, the DSP 63 judges whether or not Dm-1 is larger than 0 and Dm is equal to 0 or less. Since when Dm-1 is larger than 0 and Dm is 0 or less, the variation ratio of the correlation value varies from a positive value to 0 or across 0 at Dm, the correlation value at this time is a relative maximum or an approximate value of the relative maximum. Accordingly, the judgment result in the step S52 is Yes. Other than this, the result of the judgment result in the step S52 is No.

[ 0098 ]

When the judgment result is Yes in the step S52, the DSP 63 transmits a success report which represents that the subject audio data extremely resembles the original audio

data to the CPU 62 (step S53).

[ 0099 ]

On the other hand, if the judgment result in the step S51 is No or if the judgment result of the step S52 is No, the DSP 63 transmits a failure report representing that the subject audio data does not resemble the original audio data very much to the CPU 62.

[ 0100 ]

Fig. 12 shows graphs showing values calculated for samples of actual audio data with the calculation expressions used in the judgment processes in the step S51 and the step S52. Upon creating the graphs, a one-stage IIR (Infinite Impulse Response) filter is used as a high pass filter having a cut-off frequency of 25 Hz in the step S3 in Fig. 4; a combination of $k = 4410$ and $f = 1$ is used as constants in the comb filter in the step S5; and a one-stage IIR filter is used as a low pass filter having a cut-off frequency of 25 Hz in the step S6. Further, constants of $p = 0.5$ and $q = 0.8$ are used in a criterion in the step S51 in Fig. 11.

[ 0101 ]

The graph at the top of Fig. 12 shows the values of the numerator of the left side of the expression (2) and values in the expression which the denominator in the left side is moved to the right side with respect to m (abscissa). The middle graph in Fig. 12 shows values of the numerator of the left side of the expression (3) and values in the expression which the denominator in the left side is moved to the right side with respect to m. The bottom graph in Fig. 12 shows the values in the left side in the expression (4).

[ 0102 ]

When the value of m is within a domain A in Fig. 12, the value of the numerator at the left side of the expression (2) is equal to or greater than the value of the expression

which the denominator of the left side is moved to the right side and the condition of the expression (2) is met. In the domain A, when m is located in a domain B, the value of the numerator of the left side of the Expression (3) is equal to or greater than the value of the expression which the denominator of the left side of the Expression (3) is moved to the right side and the condition of the Expression (3) is met. As a result, Yes is acquired in the judgment process in the step S51. When the value of m is equal to a value as indicated with an arrow C in the domain B, the value of the left side of the expression (4) turns from a positive value to 0 and the condition of the Expression (5) is met; Yes is acquired in the judgment process in the step S52.

The above is the explanation on the correlation discrimination process.

[ 0103 ]

In the step S35 in Fig. 10, the DSP 63 transmits the success report or the failure report to the CPU 62 as a result of the correlation discrimination process as mentioned the above. The CPU 62 receives either of the reports from the DSP 63 and judges whether the report is a success report or a failure report (step S36).

[ 0104 ]

Normally, the correlation discrimination process meets all of the conditions expressed by the expressions (2) to (5) only when two pairs of audio data corresponding to the same portion of the musical tune are used and the CPU 62 receives the success report as a result of this. When the step S35 is executed for the first time, the original audio data used for the correlation discrimination process is the reference processed audio data (7634297) and the subject audio data is the reference processed audio data (6752298) so that the CPU 62 usually receives the failure report as a result of the step S35. In this case, the CPU 62 decrements the value of j by 1 (step S37) and returns to the process of the step

S35 mentioned above.

[ 0105 ]

When there is no audio waveform extremely similar to the audio waveform corresponding the original audio data for the last 20 seconds of the audio data NA, a series of processes of the step S35, steps S36 and step S37 are repeated 881999 times, and the process of the step S35 is repeated 882000 times. During this, the value of i is constant and the original audio data does not vary, however, the subject audio data varies due to the decrement of the value of j. For example, when the original audio data is the reference processed audio data (7634297), the subject audio data varies such as the reference processed audio data (6752298), the reference processed audio data (6752299), the reference processed audio data (6752300), ... Since it is j=0 in the 882000th step of the step S35, the subject audio data coincides with the original audio data and the CPU 62 receives the success report as a result of the step S35.

[ 0106 ]

When the CPU 62 receives the success report as a result of the step S35, the judgment result of the subsequent step S36 becomes Yes. In this case, the CPU 62 judges whether the value of j is 0 or not (step S38). As mentioned above, when there is no audio waveform extremely similar to the audio waveform corresponding to the original audio data for the last 20 seconds of the audio data NA, j is equal to 0. In this case, the CPU 62 stores the reference processed audio data (i) serving as the original audio data at that timing in the RAM 64. Further, the CPU 62 reads out the time information (i) stored in the queue of the RAM 64 in correspondence with the sample value (i) and stores it in the RAM 64 (step S39). Hereunder, the reference processed audio data stored in the RAM 64 in the process of the step S39 is called as an " processed audio data for end reference", and

the time information (i) is called as an "end time information". For example, when the judgment result of the step S38 executed for the first time becomes Yes, the end time information is the time information (7634297) corresponding to the sample value (7634297), namely, about 173.11 seconds because i is equal to 7634297.

[ 0107 ]

On the other hand, there may be an audio waveform extremely similar to the audio waveform corresponding to the original audio data for the last 20 seconds of the audio data NA. In this case, all of the conditions expressed by the expression (2) to the expression (5) in the step S35 are fulfilled before the series of processes of the above mentioned step S35, step S36 and step S37 are repeated 881999 times, and as a result, the CPU 62 receives the success report. As a result, the judgment result becomes Yes in the step S36 and it proceeds to the step S38, however, the judgment result of the step S38 becomes No since j does not become 0.

[ 0108 ]

If the judgment result in the step S38 is No, the CPU 62 modifies the reference processed audio data (i) serving as the original audio data and repeats the step S32 to step S38. The CPU 62 judges whether i = W + 65536 + 881999 = W + 947535 = 7297935 or not (step S40). If the reference raw audio data (i) used for generating the original audio data (i) happens to be the 882000th reference raw audio data from the start of the queue, the judgment result in the step S40 becomes Yes. When the step S40 is executed for the first time, the judgment result of the step S40 becomes No because i = Z = 7634297.

[ 0109 ]

When the judgment result in the step S40 is No, the CPU 62 decrements i by 1. The CPU 62 sets 881999 in j (step S41). Thereafter, the CPU 62 returns the process to

the step S32. Since i is decremented by 1 and j is equal to 881999 in the step S32, the reference raw audio data located one sample before the reference raw audio data, which the data generation process for correlation discrimination has already been executed in the preceding step S32, toward the start side in the queue is read out from the queue as the reference raw audio data (i-j). For example, since i is equal to 7634296 in the step S32 executed immediately after the process in the S41 is executed for the first time, the data generation process fro correlation discrimination is executed on a reference raw audio data (6752297). As a result, a new reference processed audio data (6752297) is stored in the RAM 64 in addition to the reference processed audio data (7634297) to the reference processed audio data (6752298) that are already stored in the RAM 64.

[ 0110 ]

Since j is equal to 881999 in the judgment process in the subsequent step S33, the judgment result becomes Yes. Subsequently, the CPU 62 repeats the series of processes of the step S35, step S36 and step S37.. The CPU 62 receives a success report as a result of the correlation discrimination process in the step S35 and the CPU 62 executes the judgment process of the step S38 when the judgment result in the step S36 becomes Yes.

[ 0111 ]

As mentioned already above, when there no audio waveform extremely similar to the audio waveform corresponding to the original audio data for the last 20 seconds of audio waveform of the audio data NA corresponding to the audio data having the sample value (i) as the end, j becomes 0. Accordingly, the CPU 62 executes the step S39 and as a result of this, stores the processed audio data for end reference and the end time information in the RAM 64. On the other hand, when there no audio waveform extremely similar to the audio waveform corresponding to the original audio data for the

last 20 seconds of audio waveform of the audio data NA corresponding to the audio data having the sample value (i) as the end, j becomes larger than 0. Accordingly, the CPU 62 executes the judgment process in the step S40.

[ 0112 ]

If the judgment result in the step S38 repeatedly becomes No, the value of i is decremented due to the repeat of the processes of the step S40 and the step S41. For example, when the audio waveform represented by the sample values stored in the queue is constant, the judgment result in the step S38 does not become Yes. As a result, it becomes i = W + 947535 = 7297935 so that the judgment result in the step S40 becomes Yes. This means that the audio waveform does not have any featured shape in the audio waveform near the end of the audio data NA acquired by the sample values stored in the queue and as a result, the processed audio data for end reference is failed to acquire. Accordingly, the CPU 62 causes the manipulation display 5 to display an error message (step S42) and finishes the series of processes.

The above is a generation process of the processed audio data for end reference.

[ 0113 ]

[ 1.2.1.11: Storing SMF in FD ]

By the process in the step S39 in the above generation process of the processed audio data for end reference, the CPU 62 subsequently executes a recording process of the SMF into the FD when the processed audio data for end reference and the end time information are stored in the RAM 64.

[ 0114 ]

First, the CPU 62 reads out the following data stored in the RAM 64.

(1) processed audio data for start reference

(2) start time information

(3) event data

(4) processed audio data for end reference

(5) end time information

[ 0115 ]

Subsequently, the CPU 62 uses the readout data and generates a track chunk of the SMF. Further, the CPU 62 attaches the header chunk to the generated track chunk in order to create the SMF.

[ 0116 ]

Fig. 13 is a view to show the general overview of the SMF generated by the CPU 62. A system exclusive event including the processed audio data for start reference and the start time information and a system exclusive event including the processed audio data for end reference and the end time information are stored at the start of the data area of the track chunk together with the corresponding delta time, respectively. The delta time for the system exclusive events are arbitrary and it is assumed to be 0.00 second in the present embodiment. Following the above mentioned system exclusive events, the management event and the performance event are stored together with the corresponding delta time in the order of the delta time.

[ 0117 ]

The CPU 62 finishes creating the SMF and transmits the created SMF to the FD drive 2 together with a writing instruction. The FD drive 2 receives the write instruction and the SMF from the CPU 62 and writes the SMF in the loaded FD.

[ 0118 ]

Fig. 14 is a view to show the audio data NA and the time information and the

delta time written in the SMF. The start time information stored in the SMF together with the processed audio data for start reference represents the time when the musical tune N actually starts in the audio data NA. The end time information stored in the SMF together with the processed audio data for end reference represents the time when the musical tune N actually ends in the audio data NA.

[0119]

[ 1.2.2: Playback operation ]

Subsequently, the operations to synchronously play back the audio data included in the music CD and the MIDI data in the SMF by using the SMF recorded by the above mentioned method will be explained. The music CD used during the playback operation below includes the audio data of the musical tune N used during the above mentioned recording operation, however, the version is different and a time period from the playback start of the music data until the musical tune N actually starts and the level of the audio waveform represented by the audio data are different. Since audio effects on the audio data are edited when the data for the music CD is created from the master data of the musical tune N, the audio data of the musical tune N included in the music CD is slightly different from the audio data NA included in the music CD-A. Hereunder, the music CD used for the playback operation, which will be explained hereunder, is called as a music CD-B and the audio data included in the musical tune N in the music CD-B is called as an audio data NB.

[0120]

[ 1.2.2.1: Playback start manipulation ]

The user loads a music CD-B on the CD drive 1 and an FD, on which the SMF is recorded, on the FD drive 2. Subsequently, the user depresses the key pad of the

manipulation display 5 corresponding to the synchronized playback start of the audio data NB and the SMF. The manipulation display 5 outputs a signal corresponding to the depressed key pad to the controller 6.

[ 0121 ]

The CPU 62 receives a signal instructing the synchronized playback start from the manipulation display 5 and transmits a transmission instruction of the SMF to the FD drive 2. The FD drive 2 reads out the SMF from the FD in response to the transmission instruction of the SMF and transmits the readout SMF to the controller 6. The CPU 62 receives the SMF from the FD drive 2 and stores the received SMF in the RAM 64.

[ 0122 ]

[ 1.2.2.2: Timing adjustment of performance event ]

The CPU 62 stores the SMF in the RAM 64 and adjusts the timings of the performance events. The timing adjustment process of the performance event is a process of adjusting a time shift between the playback of the musical tune N in the audio data NB and the playback of the MIDI data of the SMF, which is generated by differences of the silent periods before the start of the musical tune and the silent periods after the musical tune of the audio data NA used for recording the SMF and the audio data NB used for playback of the SMF, namely, a difference between the start timings of the musical tune N and the playback speeds of the musical tune N. Hereunder, the timing adjustment process of the performance event will be explained with reference to Fig. 15.

[ 0123 ]

The CPU 62 creates a counter i and sets 65535 in i (step S61). Subsequently, the CPU 62 sequentially transmits a playback instruction of the audio data NB to the CD drive 1 and the CD drive 1 transmits sample values of the audio data NB from the start to the

controller 6 for every 1/44100 second. In the following explanation, the sample value of the audio data NB is expressed as sample value (0), sample value (1), ... from the start as similar to the audio data NA.

[ 0124 ]

The CPU 62 receives a first sample value of the audio data NB, namely, the sample value (0), from the CD drive 1 and starts time measurement from the timing (hereinafter called as "a base timing Q") based on clock signals acquired from the clock.

[ 0125 ]

The CPU 62 sequentially stores the receives sample values in the queue of the RAM 64 together with the time information representing the receiving timing. The time information is a lapse of time from the base timing Q. Hereinafter, the time information for the sample value (n) is called as the time information (n) as similar to the above mentioned recording operation. It is assumed that the number of the sample values recordable in the queue is 1323000 pairs.

[ 0126 ]

The CPU 62 stores the sample value (i) in the queue and transmits an execution instruction of the data generation process for the correlation discrimination on the 65536 pairs of sample values having the sample value (i) stored in the queue as the last one to the DSP 63. Hereinafter, the series of 65536 pairs of sample values having the sample value (n) as the last one is called as the reference raw audio data (n) as similar to that in the above mentioned recording operation. The DSP 63 executes the data generation process for correlation discrimination on the reference raw audio data (i) (step S62). Since the data generation process for the correlation discrimination in the step S62 is similar to the process, which has been explained already with reference to Fig. 4, the explanation thereof

will be omitted. Since i is equal to 65535 when the step S62 is executed for the first time, the data generation process for correlation discrimination is executed on the reference raw audio data (65535). As a result of the process in the step S62, the reference processed audio data (i) is stored in the RAM 64.

[ 0127 ]

Subsequently, the CPU 62 transmits an execution instruction of the correlation discrimination process on the reference processed audio data (i) of the processed audio data for start reference to the DSP 63. The DSP 63 reads out the processed audio data for start reference included in the SMF stored in the RAM 64 and the reference processed audio data (i) stored in the RAM 64 in the step S62 and executes the correlation discrimination process (step S63). Since the correlation discrimination process in the step S63 is similar to the process which has been explained already with reference to Fig. 11, the explanation will be omitted. As a result of the process in the step S63, the DSP 63 transmits a success report or a failure report to the CPU 62.

[ 0128 ]

The CPU 62 receives either of the failure report and the success report from the DSP 63 and judges whether the received report is the success report or not (step S64). Since the musical tune N rarely starts at the start of the audio data NB, the judgment result normally becomes No when the step S64 is executed for the first time. In this case, the CPU 62 judges whether i is equal to 65535 + 882000 = 947535 or not (step S65). The judgment process in the step S65 is a judgment process for stopping repeat of the series of processes from the step S62 to the step S65 when the correlation discrimination process in the step S63 is failed for all of the reference processed audio data (i) acquired at the start of the audio data NB for 20 seconds. When the step S65 is executed for the first time, the

judgment result becomes No since i is equal to 65535. The CPU 62 increments the value of i by 1 (step S66) and moves to the process in the step S62.

[ 0129 ]

Subsequently, the CPU 62 repeats the series of processes from the step S62 to the step S65 and the correlation discrimination process is executed by sequentially changing the subject audio data such that the subject processed audio data (65535), the subject processed audio data (65536), the subject processed audio data (65537), ... for the processed audio data for start reference as the original audio data. As a result, the reference raw audio data (i) used in the step S62 at several times reaches the audio data representing the part of the musical tune N corresponding to the processed audio data for start reference.

[ 0130 ]

For example, it is assumed that a part corresponding to the processed audio data for start reference for the musical tune N are the series of audio data having the sample value (28740) in the audio data NB as a start, and the reference raw audio data (94275), which is a subject of the data generation process for correlation discrimination in the step S62 for the 28741st times, after repeating the series of processes from the step S62 to the step S66 for 28740 times is a part corresponding to the processed audio data for start reference for the musical tune N in the audio data NB.

[ 0131 ]

As described the above, when the reference raw audio data (i) used in the step S62 shows the part of the musical tune N corresponding to the processed audio data for start reference, the reference processed audio data (i) generated by the step S62 and the processed audio data for start reference are audio data generated by applying the same data

generation process for correlation discrimination on the audio data showing the same part of the musical tune N. Accordingly, the similarity of the audio data for these audio data is extremely high and the CPU 62 receives a success report as a result of the correlation discrimination process in the step S63. As a result, the judgment result of the step S64 becomes Yes.

[ 0132 ]

When the judgment result in the step S64 is Yes, the CPU 62 transmits a stop instruction of the playback of the audio data NB to the CD drive 1 and the CD drive 1 stops the transmission of the audio data NB. Subsequently, the CPU 62 calculates a value of (i - 65535) / 44100 and also calculates the time information corresponding to the start of the reference processed audio data (i). For example, it is assumed that i is equal to 94275 when the judgment result in the step S64 becomes Yes, the time information corresponding to the start of the reference processed audio data (i) is about 0.65 second because (94275 - 65535)/44100 is about 0.65. Subsequently, the CPU 62 reads out the start time information from the SMF stored in the RAM 64 and calculates the time difference by using the time information corresponding to the start of the reference processed audio data (i) and the readout start time information. Hereunder, the time difference is called as a "top offset". The top offset becomes negative when the starting timing of the musical tune N of the audio data NB is earlier than the musical tune N of the audio data NA and becomes positive when it is later than that. For example, it is assumed that the time information is 0.65 second corresponding to the start of the reference processed audio data (i) and the start time information is 1.15 seconds, the top offset is -0.50 second because 0.65 - 1.15 is equal to -0.50. The CPU 62 stores the top offset in the RAM 64 (step S67).

[ 0133 ]

The CPU 62 reads out the start time information and the end time information from the SMF stored in the RAM 64 after recording the top offset in the step S67 and calculates the time difference. Subsequently, the CPU 62 multiplies 44100 to the time represented by the time difference and calculates the number sample values corresponding to the time difference. The number of the sample values are the number of sample values corresponding to the time from the start of the processed audio data for start reference and the end of the processed audio data for end reference. The CPU 62 subtracts 65536 from the number of the sample values and calculates the number of the sample value corresponding to the time period from the end of the processed audio data for start reference to the end of the processed audio data for end reference. The number of the sample values calculated in such a way is defined as "V".

[ 0134 ]

For example, it is assumed that the start time information is 1.15 seconds and the end time information is 173.11 seconds, the time period from the start of the processed audio data for start reference to the end of the processed audio data for end reference is 173.11 - 1.15 = 171.96 (seconds). The number of sample values corresponding the time period is 171.96 X 44100 = 7583436. Accordingly, the number of the sample values corresponding to the time period from the end of the processed audio data for start reference and the end of the processed audio data for end reference, namely, V is 7583436 - 65536 = 7517900.

[ 0135 ]

Subsequently, the CPU 62 creates a counter j and set j = i + V – 441000 (step S68). In the audio data NB, the reference raw audio data (i) is the audio data corresponding to the start portion of the musical tune N for about 1.49 seconds.

Accordingly, the audio data corresponding to the end portion of the musical tune N for about 1.49 seconds is estimated to locate around the reference raw audio data ( i + V). Then, the reference raw audio data (j), namely, the reference raw audio data (i + V – 441000) is the data locating 10 seconds before the reference raw audio data ( i+ V).

[ 0136 ]

Subsequently, the CPU 62 transmits a playback instruction for the sample value (j-65535) or after of the audio data NB to the CD drive 1 and the CD drive 1 sequentially transmits the sample value (j-65535), the sample value (j-65534), ... to the controller 6. The CPU 62 sequentially stores the received sample values in the queue of the RAM 64 together with the time information representing the received time.

[ 0137 ]

The CPU 62 stores the sample value (j) in the queue and transmits an execution instruction of the data generation process for correlation discrimination on the reference raw audio data (j) to the DSP 63. The DSP 63 executes the data generation process for correlation discrimination on the reference raw audio data (i) (step S69). Since the data generation process for correlation process in the step S69 is similar to the process which was already explained with reference to Fig. 4, the explanation will be omitted. As a result of the process in the step S69, the reference processed audio data (j) is stored in the RAM 64.

[ 0138 ]

Subsequently, the CPU 62 transmits an execution instruction of the correlation discrimination process of the process audio data for end reference with respect to the reference processed audio data (j). The DSP 63 reads out the processed audio data for end reference included in the SMF stored in the RAM 64 and the reference processed

audio data (j) stored in the RAM 64 in the step S69 and executes the correlation discrimination process (step S70). Since the correlation discrimination process in the step S70 is similar to the process which has been already explained with reference to Fig. 11, the explanation will be omitted. As a result of the process in the step S70, the DSP 63 transmits a failure report or a success report to the CPU 62.

[ 0139 ]

The CPU 62 receives the failure report or the success report from the DSP 63 and judges whether the received report is the success report or the failure report (step S71). The time period from the start to the end of the musical tune N in the audio data NB is rarely shifted from the time period from the start to the end of the musical tune N of the audio data NA by 10 seconds so that the judgment result is normally No when the step S71 is executed for the first time. In this case, the CPU 62 increments j by 1 (step S72) and transmits the execution instruction of the data generation process for the correlation discrimination on a next reference raw audio data to the DSP 63. Herein, if the value of j is larger than the total number of the sample values of the audio data NB, namely, when the reference raw audio data (j-1) already reaches the end of the audio data NB, the DSP 63 fails to read out the reference raw audio data (j) from the queue and transmits an error report to the CPU 62 (step S73). When the step S72 is executed for the first time, the sample value (j) does not reach the last sample value of the audio data NB yet so that the CPU 62 executes the data generation process for the correlation discrimination on a new reference raw audio data (j) without receiving the error report from the DSP 63 in the step S73 (step S69).

[ 0140 ]

After that, the CPU 62 repeats the series of processes of the step S69 to step S73,

and sequentially changes the reference processed audio data (j) as the subject audio data for the processed audio data for end reference as the original audio data in order to execute the correlation discrimination process. As a result, the reference raw audio data (j) normally used in a several time occasion of the step S69 reaches the audio data representing a part of the musical tune N corresponding to the processed audio data for end reference. As a result, the reference processed audio data (j) generated in the step S69 and the processed audio data for end reference are the sets of audio data generated by executing the same data generation process for correlation discrimination on the sets of audio data representing the same part of the musical tune N. Accordingly, the similarity of the sets of audio data are extremely high and the CPU 62 receives a success report as a result of the correlation discrimination process in the step S70. As a result, the judgment result in the step S71 is Yes.

[ 0141 ]

When the judgment result in the steps S71 is Yes, the CPU 62 transmits a stop instruction of the playback of the audio data NB to the CD drive 1 and the CD drive 1 stops transmitting the audio data NB. Subsequently, the CPU 62 calculates the value of j / 44100 in order to calculate the time information corresponding to the end of the reference processed audio data (j). For example, it is assumed that i = 7651790 when the judgment result in the step S71 becomes Yes, 7651790 / 44100 is equal to about 173.51 and the time information corresponding to the end of the reference processed audio data (j) is about 173.51 seconds. Subsequently, the CPU 62 reads out the end time information from the SMF stored in the RAM 64 and calculates the time difference by using the time information corresponding to the end of the reference processed audio data (j) and the readout end time information. Hereunder, the time difference is called as an “end offset”.

The end offset becomes negative or positive when the end point of the musical tune N of the audio data NB is earlier or later than the musical tune N of the audio data NA, respectively. For example, it is assumed that the time information corresponding to the end of the reference processed audio data (j) is about 173.51 seconds and the end time information is 173.11 seconds, the top offset is 0.40 second since 173.51 – 173.11 is equal to 0.40. The CPU 62 stores the end offset in the RAM 64 (step S74).

[ 0142 ]

The CPU 62 finishes storing the end offset in the step S74 and adds a system exclusive event including the top offset and the end offset to the SMF stored in the RAM 64 (step S75). Fig. 16 is a view to show the content of the SMF after adding the top offset and the end offset. The delta time for the system exclusive event including the top offset and the end offset is arbitrary and it is assumed to be 0.00 second in the present embodiment.

[ 0143 ]

Subsequently, the CPU 62 reads out the whole data relating to the performance event from the SMF and calculates the adjusted delta time based on an expression (6) using the delta time and stores the result in the RAM 64 (step S76). However, it is assumed that d is the delta time after adjustment; D is the delta time before the adjustment, $N_T$ is the start time information; $N_E$ is the end time information; $O_T$ is the top offset; and $O_E$ is the end offset in the expression (6).

[ Expression 6 ]

$$d = (N_T + O_T) + (D - N_T) \times \frac{(N_E + O_E) - (N_T + O_T)}{(N_E - N_T)} \quad \text{---} \quad (6)$$

[ 0144 ]

In the expression (6), the first term ($N_T$+ $O_T$) represents a start timing of the musical tune N with respect to a playback timing of the first sample value of the audio data NB in the audio data NB. (D – $N_T$) represents an execution timing of the performance event with respect to the start timing of the musical tune N in the audio data NA. ($N_E$ + $O_E$) - ($N_T$ + $O_T$) represents the time period of the whole musical tune N in the audio data NB and ($N_E$ - $N_T$) represents the time period of the whole musical tune N in the audio data NA. Accordingly, the second term represents an execution timing of the execution timing of the performance event with respect to the start timing of the musical tune N in the musical tune NB. From the above, the sum of the first term and the second term, namely, d represents an execution timing of the performance event with respect to on the playback timing of the first sample value of the audio data NB.

[ 0145 ]

For example, when the data in the SMF illustrated in Fig. 16 is used, D = 2.11, $N_T$ = 1.15, $N_E$ = 173, $O_T$ = - 0.50, $O_E$ = 0.40 for the first performance event, the delta time after adjustment corresponding to the performance event is about 1.62 seconds.

[ 0146 ]

The CPU 62 finishes the adjustment process of the delta time with respect to the performance event in the step S76 and plays back the audio data NB and the MIDI data (step S77). First, the CPU 62 transmits a playback instruction of the audio data NB to the CD drive 1. The CD drive 1 sequentially transmits sample values of the audio data NB to the controller 6 in response to the playback instruction for every 1 / 44100 second. The CPU 62 receives a first sample value of the audio data NB, namely, a sample value (0), from the CD drive 1 and starts time measurement with respect to that timing (hereinafter

referred to as "base timing R") based on clock signals from the clock.

[ 0147 ]

After that, the CPU 62 sequentially receives the sample value (1), the sample value (2), ... after the sample (0). The CPU 62 sequentially transmits the received sample values to the tone generating portion 4. The tone generating portion 4 receives the sample values and converts them into sounds for output. As a result, the user can listen to the musical tune N.

[ 0148 ]

The CPU 62 executes the transmission process of the sample values to the tone generating portion 4 and, at the same time, sequentially compares the adjusted delta time of the performance event stored in the RAM 64 and the time measurement result with respect to the base timing R in order to transmit the performance event corresponding to the delta time, which the time measurement result coincides with the delta time, to the automatic performance piano 3.

[ 0149 ]

In the automatic player piano 3, the MIDI event control circuit 34 receives the performance events from the CPU 62 and transmits the received performance events to the tone generator 35 or the driving portion 36. When the performance events are transmitted to the tone generator 35, the tone generator 35 sequentially transmits audio data representing the sounds of the musical instrument to the tone generating portion 4 in accordance with the received performance events. The tone generating portion 4 outputs the performance by the musical instrument tone received from the tone generator 35 from the speaker 44 together with the sounds of the musical tune of the audio data NB which the playback has already been started. On the other hand, when the performance events are

transmitted to the driving portion 36, the driving portion 36 moves the keys and pedals of the piano 31 in accordance with the received performance events. In either case, the user can listen to the musical tune N stored in the audio data NB and the performance with the musical instrument tone by the performance information stored in the SMF simultaneously. After that, the CD drive 1 finishes the transmission of the last sample value of the audio data NB and stops the playback of the audio data NB so that the playback of the audio data NB and the MIDI data is finished.

[ 0150 ]

The CPU 62 receives the last sample value of the audio data NB and finishes the process in the step S77, and transmits a display instruction of a message to urge the user to save the top offset and the end offset or not to the manipulation display 5. The manipulation display 5 displays a message in response to the display instruction (step S78). The user depresses a key pad corresponding to "save" on the manipulation display 5 in response to the message of the step S78 and the CPU 62 reads out the SMF shown in Fig. 16 from the RAM 64 in response to the signal received from the manipulation display 5 and transmits the readout SMF to the FD drive 2 together with a save instruction . The FD drive 2 receives the SMF and the save instruction and overwrites the SMF saved in the FD by the newly received SMF (step S79). After saving the SMF by the FD drive 2, the CPU 62 finishes the timing adjustment process of the series of performance events. On the other hand, when the user depresses a keypad corresponding to "not save" with respect to the message in the step S78, the CPU 62 does not execute the process in the step S79 in response to the received signal from the manipulation display 5 and finishes the timing adjustment process for the series of the performance events.

[ 0151 ]

The above mentioned process is executed when both of the correlation discrimination processes in the step S63 and the step S70 are succeeded and the correlation discrimination process may not be succeeded due to the fact the contents of the audio data NA and the audio data NB are different from each other very much. A process for that case will be explained hereunder.

[ 0152 ]

First, when the correlation discrimination result in the step S63 are failed for all of the reference processed audio data (i) acquired for the period of starting 20 seconds of the audio data NB, the judgment result in the step S65 becomes Yes. In this case, the CPU 62 executes the adjusting process manually done by the user (step S80). Hereunder, the manual adjusting process will be explained with reference to Fig. 17.

[ 0153 ]

First, the CPU 62 generates $O_T$ and $O_E$ as parameters for storing the top offset and the end offset, respectively, and sets 0 in these. Subsequently, the CPU 62 transmits a stop instruction of the playback of the audio data NB and a playback instruction from the start of the audio data NB to the CD drive 1. The CD drive 1 sequentially transmits the sample values of the audio data NB to the controller 6 for every 1 / 44100 second in response to the playback instruction. The CPU 62 receives a first sample value of the audio data NB from the CD drive 1 and starts the time measurement from that timing based on the clock signals acquired from the clock. After that, the CPU 62 sequentially transmits the received sample values to the tone generating portion 4. As a result, the user can listen to the musical tune N. The CPU 62 transmits the sample values to the tone generating portion 4 and at the same time, transmits the performance event to the automatic player piano 3 based on the delta time of the performance events of the SMF

stored in the RAM 64 and the measured time. As a result, the user can listen to the musical tune N and at the same time, the performance by the musical tones in accordance with the MIDI data (step S91).

[ 0154 ]

The CPU 62 plays back the audio data NB and the MIDI data in the step S91 and transmits a display instruction of a message urging the user to adjust the top offset to the manipulation display 5. The user follows the message displayed on the manipulation display 5 and depresses a key pad corresponding to "-" when the performance by the MIDI data is felt to be earlier than the musical tune N or a key pad corresponding to "+" when the performance by the MIDI data is felt to be later than the musical tune N. The CPU 62 increments the $O_T$ by 1/75 when the key pad corresponding to "-" is depressed and decrements $O_T$ by 1/75 when the key pad corresponding to "+" is depressed (step S92). 1/75 is a value representing a time period of a single frame (second) in the musical CD.

[ 0155 ]

The CPU 62 renews the value of $O_T$ in the step S92 and calculates the adjusted delta time by the delta time corresponding to the performance event stored in the SMF in the RAM 64 in accordance with the above mentioned expression (6). The adjusted delta time is stored in the RAM 64. Thereafter, the CPU 62 transmits the performance events to the automatic player piano 3 based on the adjusted delta time. As a result, the performance by the MIDI data is adjusted along the time line in accordance with the manipulation by the user. Accordingly, the user listens to the performance by the MIDI data reflecting the adjustment of the top offset together with the musical tune N at the same time and rapidly recognizes whether the adjustment is proper or not. The user can repeatedly execute the adjustment manipulation in the step S92 until the key pad

corresponding to "end" is depressed (step S93: No).

[ 0156 ]

The user feels that the performance by the MIDI data reflecting the adjustment of the top offset is synchronized with the musical tune N and depresses a key pad corresponding to "end". When the key pad corresponding to "end" (step S93: Yes) is depressed, the CPU 62 transmits a display instruction of a message urging the user to adjust the end offset to the manipulation display 5. The user follows the message displayed by the manipulation display 5 and depresses a key pad corresponding to "-" when the performance by the MIDI data becomes gradually faster with respect to the musical tune N or a key pad corresponding to "+" when the performance by the MIDI data becomes gradually slower than the musical tune N. The CPU 62 increments $O_E$ by 1/75 when the key pad corresponding to "-" is depressed and decrements $O_E$ by 1/75 when the key pad corresponding to "+" is depressed (step S94).

[ 0157 ]

The CPU 62 renews the value of $O_E$ in the step S94 and calculates the adjusted delta time by using the delta time corresponding to the performance event stored in the SMF in the RAM 64 in accordance with the above mentioned expression (6). The adjusted delta time is stored in the RAM 64. The CPU 62 follows the adjusted delta time and plays back the MIDI data. As a result, the performance speed by the MIDI data is adjusted by the manipulations by the user and the user listens to the performance by the MIDI data reflecting the adjustment of the end offset together with the musical tune N at the same time and rapidly recognizes whether the adjustment is proper or not. The user can repeatedly execute the adjustment manipulation in the step S94 until the key pad corresponding to "end" is depressed (step S95: No).

[ 0158 ]

The user feels that the performance by the MIDI data reflecting the adjustment of the end offset is synchronized with the musical tune N and depresses a key pad corresponding to "end". When the key pad corresponding to "end" (step S95: Yes) is depressed, the manual adjustment process is finished.

[ 0159 ]

After finishing the manual adjustment process, the CPU 62 moves the process to the step S76 in Fig. 15 and executes the step S76 to step S79 by using the top offset and the end offset which are manually adjusted. As a result, the user can listen to the musical tune N by the audio data NB and the performance by the MIDI data synchronized with the musical tune N simultaneously.

[ 0160 ]

On the other hand, if the correlation discrimination processes in the step S70 are failed for all of the reference processed audio data (i) from a timing 10 seconds before the end of the musical tune N of the audio data NB until the end of the audio data NB, the judgment result in the step S73 becomes Yes. In this case, the CPU 62 executes the adjustment process by the management event included in the SMF (step S81). Hereunder, the adjustment process by the management event will be explained hereunder.

[ 0161 ]

First, the CPU 62 transmits a playback instruction to the CD drive 1 from the start of the audio data NB. The CD drive 1 sequentially transmits the sample values of the audio data NB from the start to the controller 6 for every 1 / 44100 second in response to the playback instruction. The CPU 62 receives a first sample value of the audio data NB from the CD drive 1 and starts the time measurement from that timing based on the clock

signals acquired from the clock.

[ 0162 ]

The CPU 62 receives the sample values from the CD drive 1 and sequentially stores the sample values in the queue of the RAM 64 together with the time information representing the received time of the sample value. At the same time, the CPU 62 judges whether either of the absolute values in the left or right of the received sample values exceeds 1000 or not. When either of the absolute values in the left or right of the received sample values exceeds 1000, the CPU 62 transmits an execution instruction of the management event generation process to the DSP 63 and the DSP 63 executes the management event generation process. The management event generation process is the same as the process which has been already explained with reference to Fig. 6 so that the explanation will be omitted.

[ 0163 ]

The CPU 62 receives the management event from the DSP 63 by the management event generation process. The CPU 62 receives the management event and stores it in the RAM 64 together with the time information representing the receiving time. Hereunder, the management event included in the SMF stored in the RAM 64 is called as a management event A and the management event received by the CPU 62 by the management event generation process on the audio data NB is called as a management event B.

[ 0164 ]

After finishing the playback of the audio data NB by the CD drive 1 and storing the time information of the last management event B in the RAM 64, the CPU 62 uses the delta time of the management event A and the time information of the management event

B in order to correlate to the management event A and the management event B. Hereunder, the process of correlating the management event A and the management event B will be explained by using the data example.

[ 0165 ]

Fig. 18 is a table to show a data example of the delta time corresponding to the management events A and the time information stored in association with the management events B. However, the value at the start of the column of the audio data NA represents the start time information and the value of the start of the column of the audio data NB represents the time information by adding the top offset stored in the RAM 64 in the step S67 to the start time information, respectively. Hereunder, the time information by adding the top offset to the start time information is called as the start time information B and the start time information relating to the audio data NA is called as the start time information A in order to discriminate it from the start time information B. The first management event A, the second management event A, ... in Fig. 18 are called as a management event A1, a management event A2, .... This is similarly applicable to the management event B.

[ 0166 ]

In Fig. 18, the start time information A corresponds to the start time information B. First, the CPU 62 calculates (management event A1 - start time information A) / (management event B1 - start time information B). The value is (1.51 - 1.15 ) / (1.01 - 0.65 ) = 1.00. Subsequently, the CPU 62 judges whether the calculated value is within a predetermined range of value or not. Hereunder, the range is assumed to be 0.97 to 1.03 as an example. Assuming that the management event A1 corresponds to the management event B1, this judgment process is a process that judges whether or not the error in time

with respect to the start time information A and the start time information B is 3% or less. In this case, the error is 0% and the CPU 62 correlates the management event A1 to the management event B1. If the calculation result is smaller than 0.97, the delta time of the management event A is earlier than the time information of the management event B1 so that the CPU 62 regards that there is no management event B corresponding to the management event A1 and executes the judgment process and the correlating process similar to the above for the management event A1 and the management event B1. On the other hand, if the above described calculation result is larger than 1.03, the delta time of the management event A1 is later than the time information of the management event B1 too much, the CPU 62 regards that there is no management event A1 corresponding to the management event B1 and the judgment process and the correlating process similar to the above are executed for the management event A1 and the management event B2. The management event A and the management B correlated last are called as the management event An and the management event Bn, respectively.

[0167]

Subsequently, the CPU 62 calculates start time information B + (management event An+1 - start time information A) X (management event Bn - start time information B) / (management event An - start time information A) and estimates the time information corresponding to the management event B based on the delta time of the management event An+1. Fig. 19 shows the estimated value of the time information of the management event B calculated from the above expression. It is assumed that the management event An is a management event A1 and the management event Bn is a management event B1, for example, and the estimated value of the management event B is 0.65 + (2.38 - 1.15) X (1.01 - 0.65) / (1.51 - 1.15) = about 1.88 (seconds). The CPU 62

judges whether a difference between the estimated value and the management event Bn+1 is within the predetermined range or not. Hereunder, this range is -0.20 to 0.20. The judgment process is a process to judge whether or not the difference between the time information of the management event B estimated from the delta time of the management event An+1 and the time represented by the time information of the management event Bn+1 is 0.20 second or less. When the difference between the estimated value and the management event Bn+1 is within the range of -0.20 to 0.20, the CPU 62 correlates the management event An+1 to the management event Bn+1.

[ 0168 ]

On the other hand, when the value calculated by subtracting the time information of the management event Bn+1 from the estimated value is smaller than -0.20, the CPU 62 judges that there is no management event B corresponding to the management event An+1 and executes the above described judging process and the correlation process by using by using the management event An instead of the management event An+1. When the value calculated by subtracting the time information of the management event Bn+1 from the estimated value is larger than 0.20, the CPU 62 judges that the management event A corresponding to the management event Bn+1 does not exist and executes the above described judging process and the correlation process by using the management event Bn+2 instead of the management event Bn+1.

[ 0169 ]

It is assumed that the management event An is a management event A5 and the management event Bn is a management event B5, and the time information of the management event B estimated based on the delta time of the management event A6 is about 8.25 seconds while the time information of the management event B6 is 9.76

seconds so that the difference is about -1.51 seconds. Accordingly, it is regarded that the management event A6 does not have the management event B. It is assumed that the management event An is a management event A9 and the management event Bn is a management event B8 and the time information of the management event B estimated from the delta time of the management event A10 is about 17.79 seconds while the time information of the management event B9 is about 15.57 seconds so that the difference is about 2.22 seconds. Accordingly, it is regarded that the there is not management event A corresponding to the management event B9.

[ 0170 ]

The CPU 62 finishes the above described correlating process on all of the management event A and the management event B and estimates the relational expression between the delta time of the management event A and the time information of the management event B by using plural pairs of the delta time of the management event A and the time information of the management event B which are correlated each other. Fig. 20 is a graph showing a regression line calculated by the least square method on the data including the data example of Fig. 19. Subsequently, the CPU 62 reads out the end time information from the SMF stored in the RAM 64 and the readout end time information is substituted with the estimated expression in order to calculate the time information corresponding to the last part of the musical tune N in the audio data NB. In an example shown in Fig. 20, for example, the relation of (time information of the management event B) = (delta time of the management event A) X 1.0053 - 0.5075 is estimated so that the 173.11 seconds serving as the end time information is substituted with the expression in order to calculate about 173.52 seconds. The time information calculated in such a way and corresponding to the last part of the musical tune N of the audio data NB is called as

an end time information B and the end time information of the audio data NA is called as an end time information A below in order to discriminate it from the end time information B.

[ 0171 ]

After calculating the end time information B, the CPU 62 calculates (end information time B - end information time A) and stored the calculated value as the end offset in the RAM 64. Subsequently, the CPU 62 adds a system exclusive event including the top offset and the end offset to the SMF stored in the RAM 64.

After that, the CPU 62 executes the step S76 to the step S79 in Fig. 15. As a result, the user can listen to the musical tune N by the audio data NB and the performance by playing back the MIDI data correctly synchronized with the musical tune N simultaneously.

[ 0172 ]

Fig. 21 is a view properly arranging the audio data NB, parts corresponding to the processed audio data for start reference and the processed audio data for end reference in the reference processed audio data generated from the audio data NB, management events generated from the audio data NB and performance events along the time axis. As shown in the bottom raw in Fig. 21, when the timing adjustment process is not executed, the performance events are transmitted to the automatic player piano 3 at 2.11 seconds, 2.62 seconds, 3.60 seconds after the start timing of the audio data NB, respectively. Since the silent period before starting the musical tune N of the audio data N is shorter than the silent period before starting the musical tune N of the audio data NA in this case, the start of the performance with the MIDI data is delayed. Since the entire time period of the musical tune N by the audio data NB is longer than the entire time period of musical tune N of the

audio data NA, the performance speed by the MIDI data is faster than the musical tune N and the performance by the MIDI data leads the musical tune N in the later part of the musical tune N. In contrast, when the timing is adjusted, the performance events are transmitted to the automatic player piano 3 at correct timings due to the timing adjustment of the performance events by the expression (6) by using the top offset and the end offset.

[ 0173 ]

[ 1.2.2.3: Synchronized playback and manual re-adjustment ]

When the user directs to save the SMF including the top offset and the end offset in an FD in the step S78 in the timing adjustment process of the performance events, the user does not need to execute the timing adjustment process of the performance events explained in Fig. 15 again when the synchronized playback of the audio data NB and the SMF are executed again. Hereunder, processes of the synchronized playback of the SMF including the top offset and the end offset and the audio data NB.

[ 0174 ]

When the user depresses a predetermined key pad on the manipulation display 5 in order to direct the synchronized playback of the SMF saved in the FD and the audio data of the audio data NB included in the music CD-B, the CPU 62 reads out the SMF from the FD through the FD drive r 2 and stored the readout SMF in the RAM 64. Subsequently, the CPU 62 reads out the top offset and the end offset of the SMF and executes the adjusting process using the expression (6) on the delta time of the performance events included in the SMF so as to store the delta time after the adjustment in the RAM 64.

[ 0175 ]

Subsequently, the CPU 62 transmits a playback instruction of the audio data NB to the CD drive 1 and the CD drive 1 sequentially transmits the sample values of the audio

data NB to the controller 6. The CPU 62 receives the first sample value and starts the time measurement from the receiving timing of the sample value. The CPU 62 sequentially transmits the received sample values to the tone generating portion 4 and the user can listen to the musical tune N. At the same time, the CPU 62 sequentially compares the delta time after the adjustment with the measured time period and transmits the performance event corresponding to the delta time to the automatic player piano 3 when the delta time after the adjustment and the measured time period match each other. As a result, the user can listen to the performance by the MIDI data in the SMF at the correct timings relative to the musical tune N.

[ 0176 ]

When the user listens to the musical tune N and the performance by the MIDI data simultaneously and feels that these are not correctly synchronized each other, the user depresses the predetermined key pad of the manipulation display 5 and the top offset and the end offset are manually adjusted. Since the manual adjustment of the tope offset and the end offset is similar to the process from the step S91 to the step S95 in Fig. 17, the explanation will be omitted. The user finishes the manual adjustment of the top offset and the end offset and depresses the predetermined key pad of the manipulation display 5 in order to save the SMF including the adjusted top offset and the end offset in the FD. The CPU 62 receives the save instruction of the SMF into the FD from the user through the manipulation display 5 and reads out the SMF including the adjusted top offset and the end offset from the RAM 64 in order to transmit the readout SMF to the FD drive 2 with the save instruction. The FD drive 2 receives the SMF and overwrites the saved SMF in the FD by the received SMF. As a result, when the user the synchronized playback of the audio data NB and the MIDI data of the SMF are synchronously played back again by the

user, the transmission timing of the performance event is adjusted by using the top offset and the end offset that are finally adjusted so that the manual adjustment does not need again.

[ 0177 ]

[ 2: Modifications ]

The above mentioned embodiments are mere illustrations of the embodiments of the present invention, and various modifications are available without departing from the feature of the present invention. Modifications will be shown hereunder.

[ 0178 ]

[ 2.1: First modification ]

In the above mentioned embodiment, the management event and the delta time of the performance event, the start time information and the end time information of the audio data NA stored in the SMF are time information acquired through the time measurement based on a timing when the CPU 62 receives the first sample value of the audio data NA. Similarly, the delta time of the management event for the audio data NB stored in the RAM 64, the time information corresponding to the start of the musical tune N in the audio data NB and the time information corresponding to the end of the musical tune N of the audio data NB are the time information acquired by time measurement by the CPU 62 based on the timing when the CPU 62 receives the first sample value of the audio data NB in the above mentioned embodiment.

[ 0179 ]

In contrast, time codes of the audio data NA and the audio data NB, which are transmitted from the CD drive 1 to the controller 6 are used instead of the time information by the measured time by the CPU 62 in the first modification. The time codes are data

stored in the music CD by corresponding to a frame as a group of audio data in the music CD, namely, 588 sample values, and each of the time codes represents a lapse of time from the starting timing of the audio data and the playback timing of the audio data corresponding to said time code.

[ 0180 ]

In the first modification, the controller 6 always transmits the clock signals to the CD drive 1. The CD drive 1 transmits the audio data to the controller 6 based on the clock signals received from the controller 6. Further, the CD drive 1 also transmits the time codes corresponding to the audio data upon transmission of the audio data. The CPU 62 stores the time information represented by the time code transmitted from the CD drive 1 with the sample value instead of the time information by the time measurement upon storing the sample value in the queue of the RAM 64. Moreover, when the time information having more accuracy needs to be handled, the CPU 62 calculates the precise time information corresponding to each of the sample values by the time code and a number of each of the sample values counted from the start of the frame corresponding to the time codes, and stores the calculated time information in the queue.

[ 0181 ]

The CPU 62 receives management event and the performance event for the audio data NA and the management event for the audio data NB and sets the time information represented by the time code corresponding to the sample value received at that timing as the time information corresponding to these event data. According to the first modification, the time codes recorded in the music CD are used and the CPU 62 does not measure the time so that the process in the CPU 62 is simplified.

[ 0182 ]

[ 2. 2: Second modification ]

In a second modification, elements of the synchronized recorder and player SS are not located in the same device and are separated into groups to be located.

For example, they are separable into following respective groups:

(1) music CD drive 1

(2) FD drive 2

(3) automatic player piano 3

(4) mixer 41 and D/A converter 42

(5) amplifier 43

(6) speaker 44

(7) manipulation display 5 and controller 6

[ 0183 ]

Further, the controller 6 may be separated into a device for recording operations only and a device for playback operations only. The element groups are connected with audio cables, MIDI cables, optical audio cables, USB (Universal Serial Bus) cables and dedicated control cables. The FD drive 2, the amplifier 43 and speakers 44 may be commercially available ones.

[ 0184 ]

According to the second modification, the location flexibility of the synchronized recorder and player SS is enhanced and the user does not need to prepare the whole new components of the synchronized recorder and player SS to reduce the cost.

[ 0185 ]

[ 2. 3: Third modification ]

In a third modification, the synchronized recorder and player SS does not include

the music CD drive 1 and the FD drive 2. On the other hand, a communication interface has a function connectable to the LAN (Local Area Network) and is connected to external communication devices through the LAN and WAN. The controller 6 has an HD (Hard Disk).

[ 0186 ]

The controller 6 receives the audio data and the audio data from other communication devices through the LAN and records the received audio data in the HD. As similarly, the controller 6 receives the SMF created in association with the audio data from other communication devices through the LAN and records the received SMF in the HD.

[ 0187 ]

The controller 6 reads out the audio data from the HD instead of receiving the audio data from the music CD drive 1. The controller 6 executes the similar operations on the HD instead of writing and reading out the SMF into or from the FD drive 2.

[ 0188 ]

According to the third modification, the user is capable of transmitting and receiving the audio data and the SMF through the LAN to the communication device which is geographically remote therefrom. The LAN may be connected to the wide area communication network such as the Internet.

[ 0189 ]

[ 2. 4: Fourth modification ]

Though all of the judgment by the absolute correlation index, the judgment by the relative correlation index and the judgment by the correlation value are used in the step S51 and the step S52 of the correlation discrimination process in the above mentioned

embodiments, the correlation discrimination process is executed by one of or plural combinations of these judgments in a fourth modification. One of or plural combinations of these judgments may be freely selectable.

According to the fourth modifications, the judgment result with the necessary accuracy is acquired with flexibility.

[ 0190 ]

[ 2. 5: Fifth modification ]

Although the relative maximum value of the correlation value is detected by the discrimination expressed by the expression (4) and the expression (5) in the step S52 of the correlation discrimination process in the above mentioned embodiments, the discrimination process expressed by the expression (4) is executed only and the relative maximum value of the correlation value is detected in the fifth modification.

[ 0191 ]

More specifically, the DSP 63 calculates a product of Dm-1 and Dm in the step S52 and judges whether or not the product is 0 or less. If the product is 0 or less, the variation ratio of the correlation value is 0 or varies across 0 so that the correlation value at that timing is a relative maximum or an approximate value of the relative maximum value. Accordingly, when the product of Dm-1 and Dm is 0 or less, the discrimination result of the step S52 becomes Yes.

[ 0192 ]

According to the fifth modification, when the local minimum value is not possible to appear around the local maximum value, the discrimination result similar to the step S52 in the above mentioned embodiments is acquired by a simpler discrimination process.

[ 0193 ]

[ 2.6: Sixth modification ]

In the above mentioned embodiment, the start time information and the end time information stored in the SMF are determined by the discrimination whether the sample value of the audio data NA exceeds a constant value or not. Accordingly, the processed audio data for start reference and the processed audio data for end reference stored in the SMF are data generated by using the audio data located around the start part and end part of the musical tune N in the audio data NA.

[ 0194 ]

However, the start time information and the end information are selected from arbitrary points in the musical tune N, respectively, in a sixth modification. For example, the user can designate suitable points based on the content of the musical tune N as the time information. A timing after lapsing a predetermined time period from the start of the audio data NA is set as the start time information and a timing a predetermined time period before the end of the audio data NA is set as the end time information.

[ 0195 ]

According to the sixth modification, it is possible to select the generation data of the processed audio data for start reference or the processed audio data for end reference by avoiding claps and cheer before and after the musical tune in a music CD recording a live performance. Even when the musical tune N has the refrain of a specific pattern around the start part or the end part thereof, the pattern is skipped and the audio data having a featured portion suitable for the correlation discrimination process is selectable as the generation data as the processed audio data for start reference and the processed audio data for end reference.

[ 0196 ]

[ 2.7: Seventh modification ]

The identification information which specifies the audio data NB is added to the SMF upon storing the SMF including the top offset and the end offset is stored in the FD in a seventh modification. The identification information specifying the audio data NB may be a combination of the identification data of the music CD-B and a track number where the audio data NB is recorded in the music CD-B. The identification data of the music CD-B is recorded as the table of content information of the music CD-B and the CPU 62 reads out the identification data from the music CD-B through the CD drive 1. The CPU 62 adds the readout identification data of the music CD-B to the SMF stored in the RAM 64 as a system exclusive event together with the track number of the audio data NB given by the instruction of the user. After that, the CPU 62 transmits the SMF stored in the RAM 64 to the FD drive 2 and the FD drive 2 saves the SMF in the FD.

[ 0197 ]

When the user loads a music CD in the CD drive 1 and designates the track number where the audio data to be played back is included is designated by the manipulation display 5, the CPU 62 first reads out the identification data of the music CD through the CD drive 1 and retrieves the SMF that stores the combination of the readout identification data of the music CD and the designated track number from the plural SMFs stored in the FD through the FD drive 2. The CPU 62 reads out the retrieved SMF and starts the synchronized playback of the audio data and the MIDI data by using the readout SMF. When the SMF storing the combination of the identification data of the music CD set in the CD drive 1 and the track number designated by the user is not stored in the FD during retrieving the SMF, the CPU 62 does not execute the synchronized playback of the audio data and the MIDI data and displays an error message on the manipulation display 5

to urge the user to set a proper FD or music CD..

[ 0198 ]

According to the seventh modification, when the user tries the synchronized playback by the combination of improper music CD and the SMF, the improper combination is instantly notified to the user so that the management of the audio data and the SMF is easily done. Further, even when plural SMFs are stored in the FD, the suitable SMF is automatically read out based on the set music CD and the designated track number so that the user does not need to specify the SMF separately.

[ 0199 ]

[ EFFECTS OF THE INVENTION ]

According to the present invention, the performance data such as the MIDI data are synchronously played back at correct timings for either of different versions of audio data which store a set of audio data having different starting timings and ending timings of the musical tune with respect to the same musical tune. Though the recording level of the musical tune may be different from each other in the different version of the same musical tune, the audio data and the performance data are simultaneously played back by the user and the starting and ending timings of the playback of the performance data are adjustable so that the playback starting timing is correctly determined for a certain version of the audio data of different recording level. Accordingly, the different performance data for the different version of the same musical tune does not need to be prepared so that the generation and management of the performance are simplified.

[ BRIEF DESCRIPTION ON DRAWINGS ]

[ Fig. 1 ] The block diagram showing the configuration of the synchronized recorder and player SS implemented by the embodiment of the present invention.

[ Fig. 2 ] The view to show the data format of SMF.

[ Fig. 3 ] The view to show the data format of MIDI event.

[ Fig. 4 ] The flowchart of the data generation process for correlation discrimination implemented by the embodiment of the present invention.

[ Fig. 5 ] The view to show the configuration of the comb filter used in the data generation process for correlation discrimination implemented by the embodiment of the present invention.

[ Fig. 6 ] The flowchart of the management event generation process implemented by the embodiment of the present invention.

[ Fig. 7 ] The view to show the configuration of the comb filter used in the management event generation process implemented by the embodiment of the present invention.

[ Fig. 8 ] The view to show the relation with respect to time between the middle term index, the long term index and the management event implemented by the embodiment of the present invention.

[ Fig. 9 ]The view to show the relation with respect to time between the management event and the performance event implemented by the embodiment of the present invention.

[ Fig. 10 ] The flowchart of the data generation process of processed audio data for end reference implemented by the present invention.

[ Fig. 11 ] The flowchart of the correlation discrimination process implemented by the embodiment of the present invention.

[ Fig. 12 ] The view to show the relation between the variation of values of calculation expressions and the discrimination result implemented by the embodiment of

the present invention.

[ Fig. 13 ] The view to show the SMF implemented by the embodiment of the present invention.

[ Fig. 14 ] The view to show the relation with respect to time of the audio data, the processed audio data for start reference, the processed audio data for end reference, the management event, the time information of the performance event and the delta time implemented by the present invention.

[ Fig. 15 ] The flowchart of the timing adjustment process of the performance event implemented by the embodiment of the present invention.

[ Fig. 16 ] The view to show the SMF implemented by the embodiment of the present invention.

[ Fig. 17 ] The flowchart of the manual adjusting process implemented by the present invention.

[ Fig. 18 ] The example of data of the delta time and the time information of the management events for the different sets of audio data implemented by the embodiment of the present invention.

[ Fig. 19 ] The data example showing the correlation process of the management event implemented by the embodiment of the present invention.

[ Fig. 20 ] The view to show relationship of the management events, delta time and time information of different sets of audio data implemented by the embodiment of the present invention.

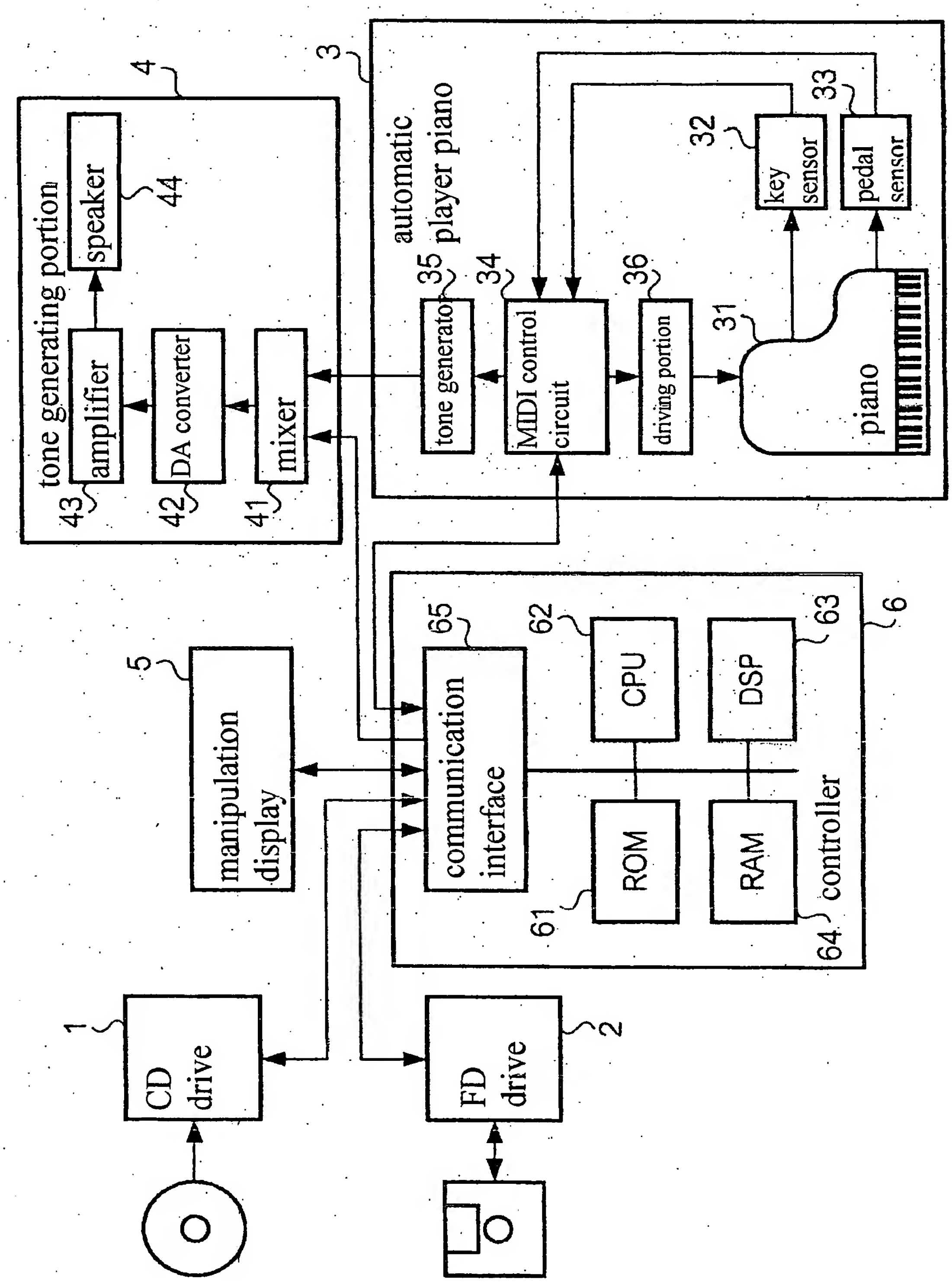
[ Fig. 21 ] The view to show the relationship with respect to time between the audio data, the reference processed audio data, management event and the time information and delta time of the performance event implemented by the embodiment of the present

invention.

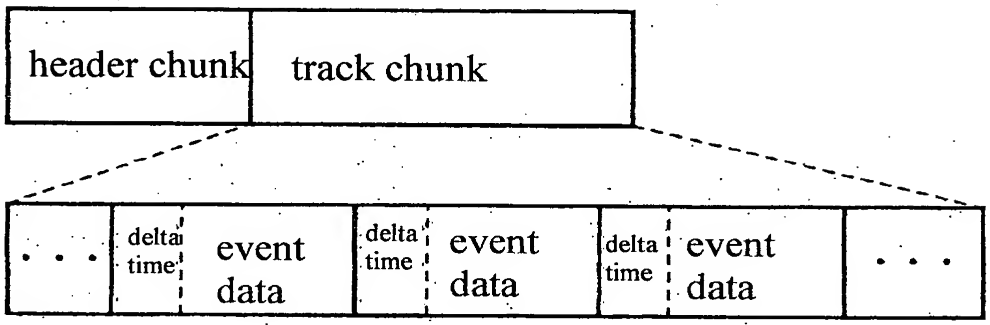
[ EXPLANATION ON REFERENCES ]

1 ... CD drive, 2 ... FD drive, 3 ... automatic player piano, 4 ... tone generating portion, 5 ... manipulation display, 6 ... controller, 31 ... piano, 32 ... key sensor, 33 ... pedal sensor, 34 ... MIDI event control circuit, 35 ... tone generator, 36 ... driving portion, 41 ... mixer, 42 ... D/A converter, 43 ... amplifier, 44 ... speaker, 61 ... ROM, 62 ... CPU, 63 ... DSP, 64 ... RAM, 65 ... communication interface

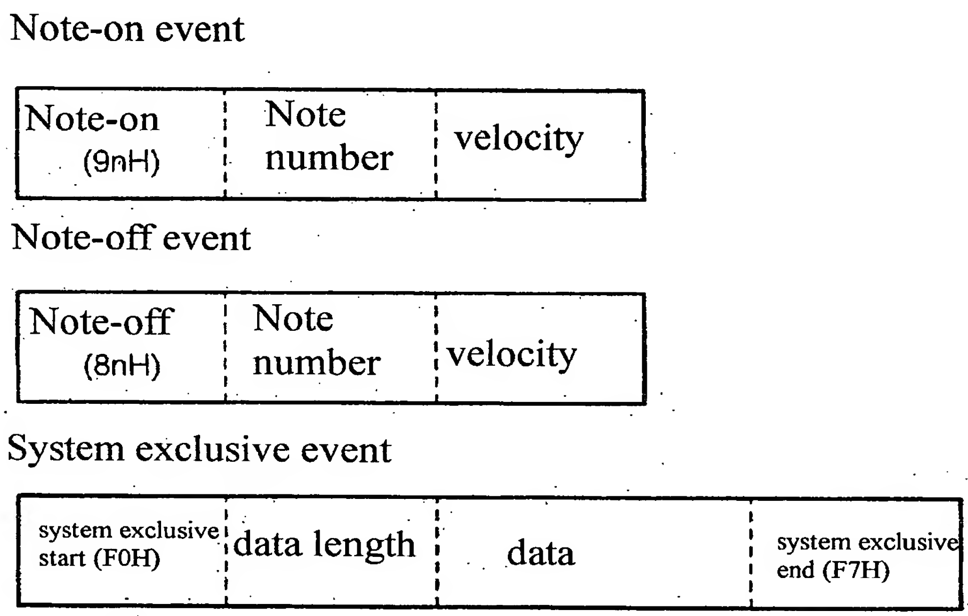
[ DOCUMENT NAME ] DRAWING
[ Fig. 1 ]



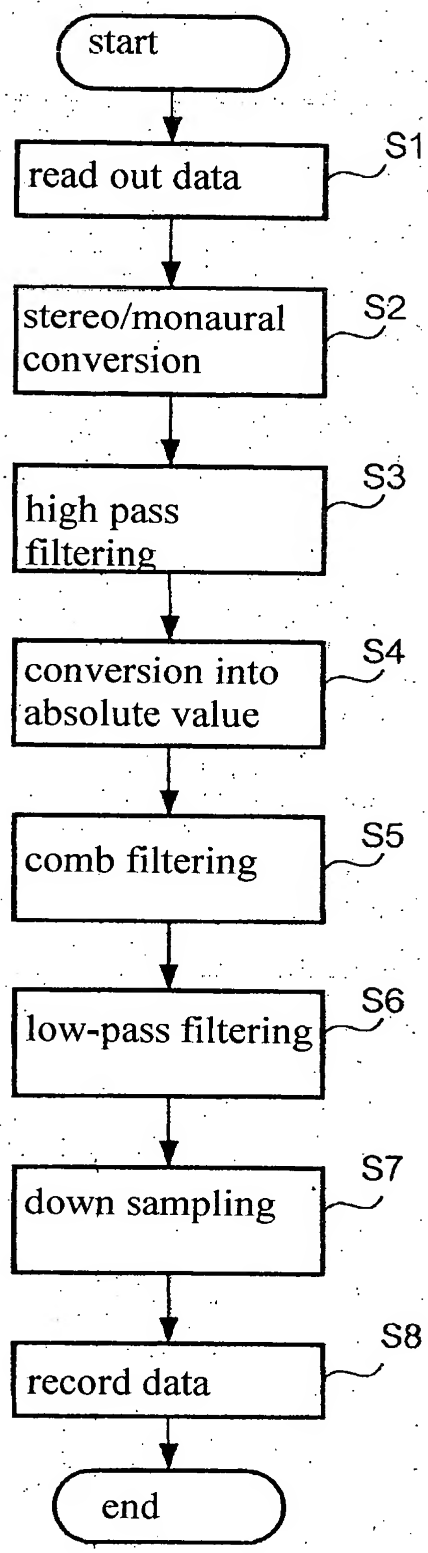
[ Fig. 2 ]


header chunk
track chunk
. . .
delta time
event data
delta time
event data
delta time
event data
. . .

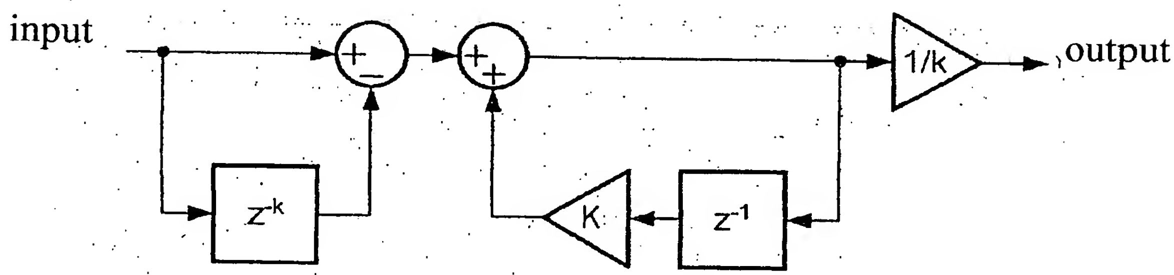
[ Fig. 3 ]


Note-on event
Note-on (9nH)
Note number
velocity
Note-off event
Note-off (8nH)
Note number
velocity
System exclusive event
system exclusive start (F0H)
data length
data
system exclusive end (F7H)

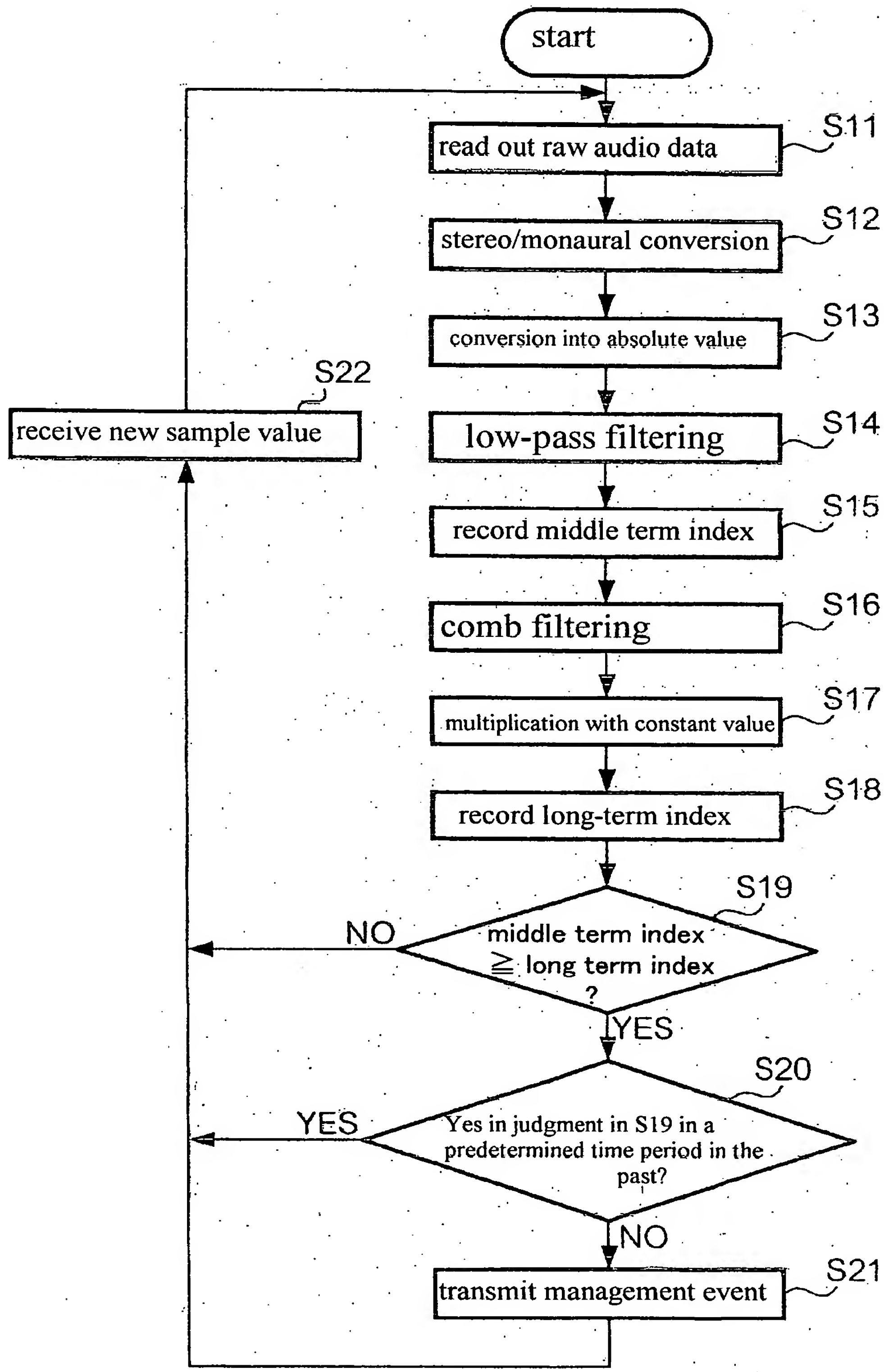
[ Fig. 4 ]


start
read out data
S1
stereo/monaural conversion
S2
high pass filtering
S3
conversion into absolute value
S4
comb filtering
S5
low-pass filtering
S6
down sampling
S7
record data
S8
end

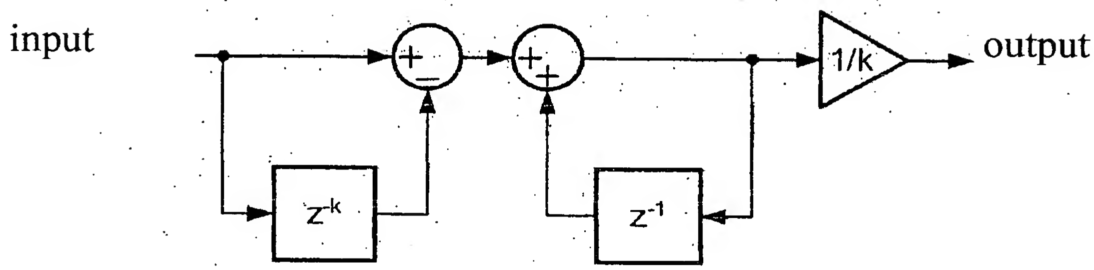
[ Fig. 5 ]



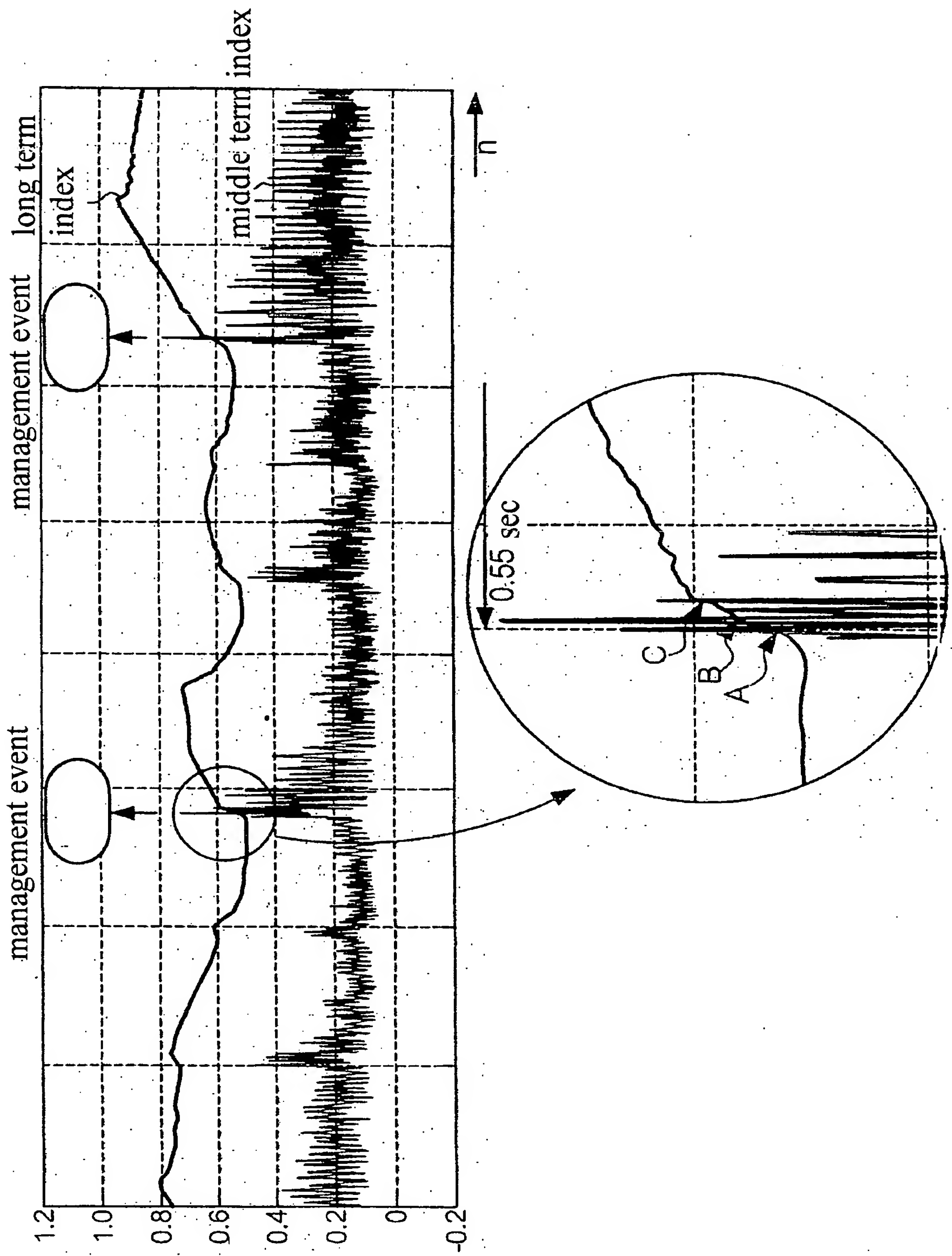
[ Fig. 6 ]


start
read out raw audio data
S11
stereo/monaural conversion
S12
conversion into absolute value
S13
low-pass filtering
S14
record middle term index
S15
comb filtering
S16
multiplication with constant value
S17
record long-term index
S18
middle term index ≧ long term index ?
S19
NO
YES
Yes in judgment in S19 in a predetermined time period in the past?
S20
YES
NO
transmit management event
S21
receive new sample value
S22

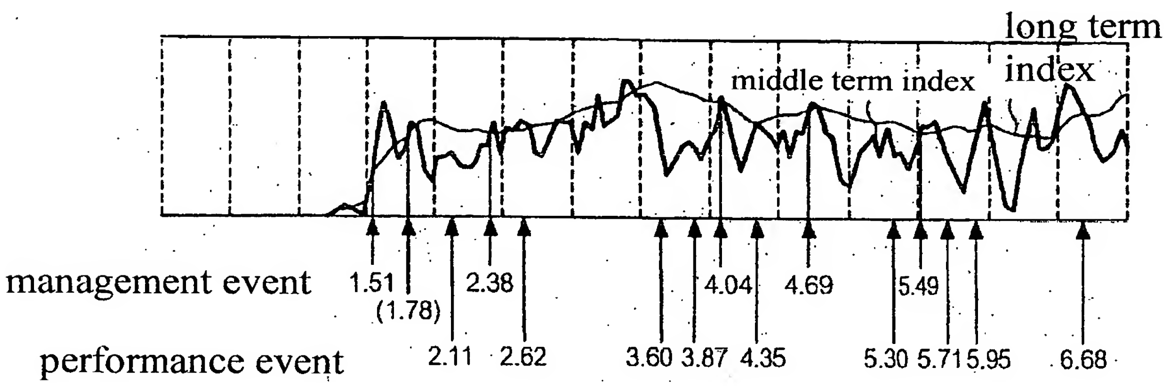
[ Fig. 7 ]



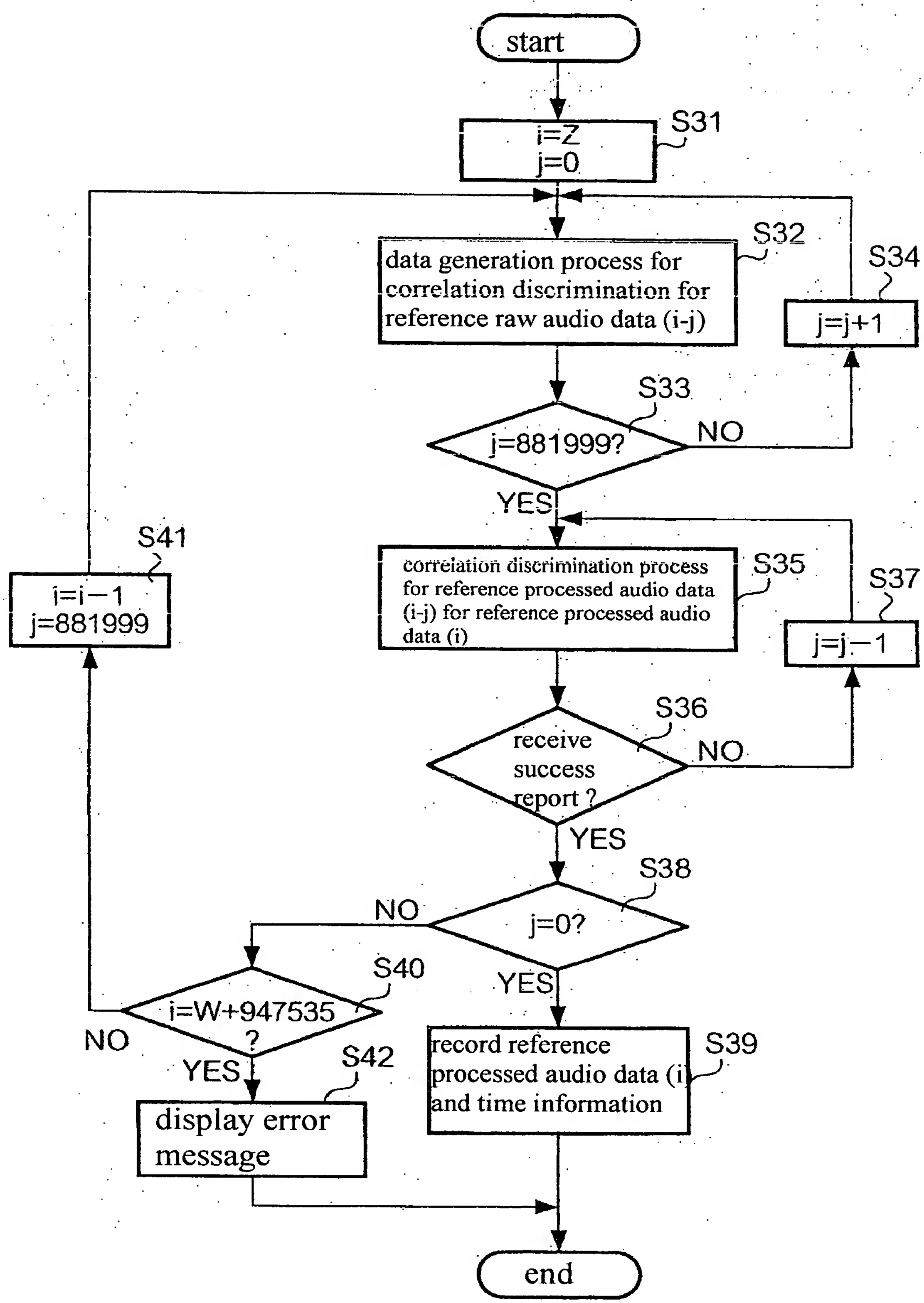
[ Fig. 8 ]


long term index
management event
management event
middle term index
n
0.55 sec
A
B
C
1.2
1.0
0.8
0.6
0.4
0.2
0
-0.2

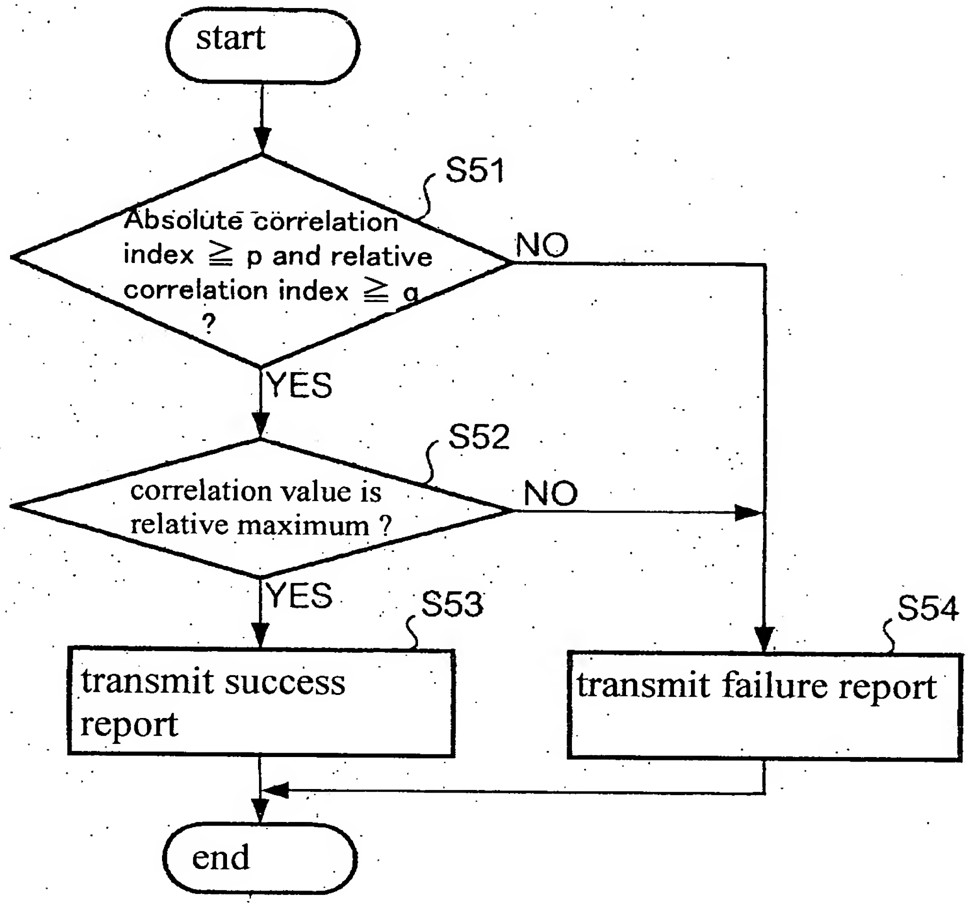
[ Fig. 9 ]


long term index
middle term index
management event
1.51
(1.78)
2.38
4.04
4.69
5.49
performance event
2.11
2.62
3.60
3.87
4.35
5.30
5.71
5.95
6.68

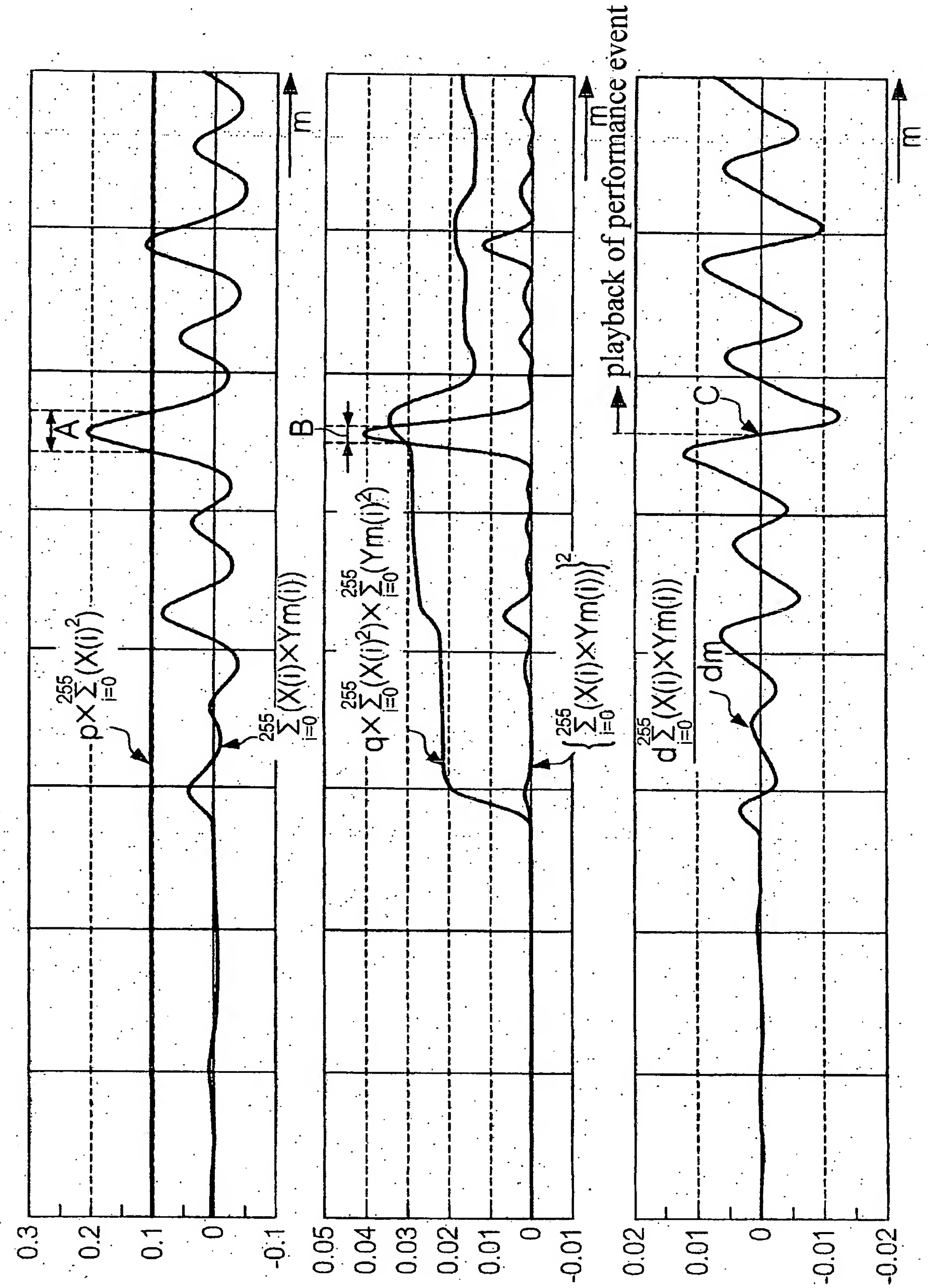
[ Fig. 10 ]


start
i=Z
j=0
S31
data generation process for correlation discrimination for reference raw audio data (i-j)
S32
S34
j=j+1
S33
j=881999?
NO
YES
S41
i=i−1
j=881999
correlation discrimination process for reference processed audio data (i-j) for reference processed audio data (i)
S35
S37
j=j−1
S36
receive success report ?
NO
YES
S38
NO
j=0?
S40
YES
i=W+947535 ?
NO
YES
S42
record reference processed audio data (i) and time information
S39
display error message
end

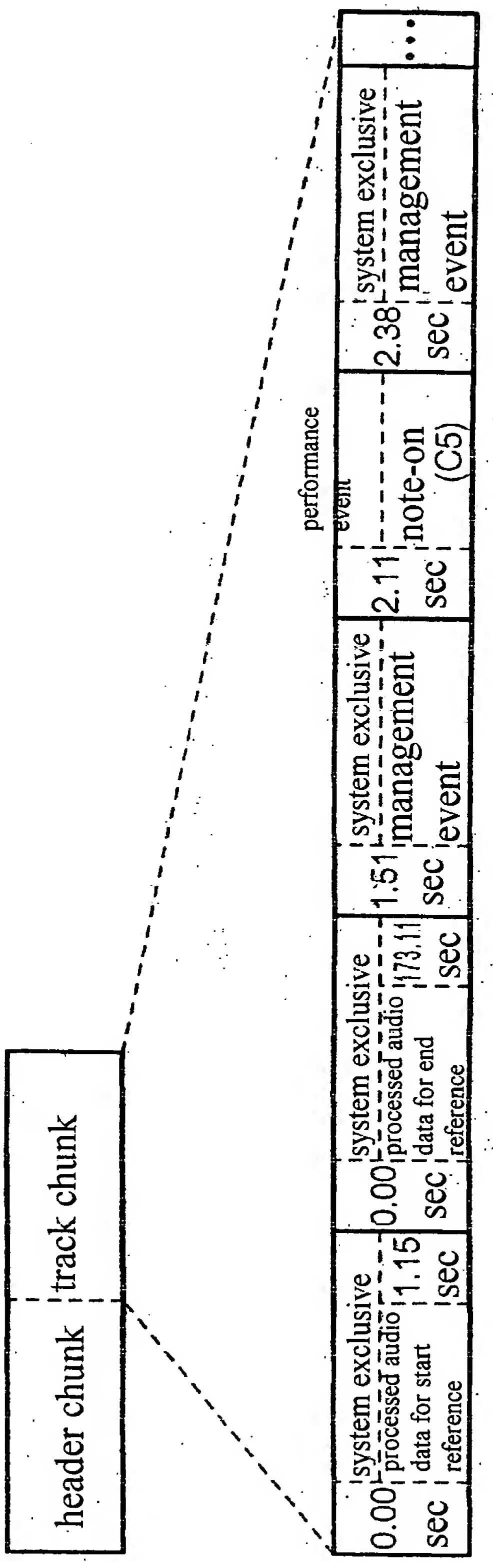
[ Fig. 11 ]


start
S51
Absolute correlation index ≧ p and relative correlation index ≧ q ?
NO
YES
S52
correlation value is relative maximum ?
NO
YES
S53
transmit success report
S54
transmit failure report
end

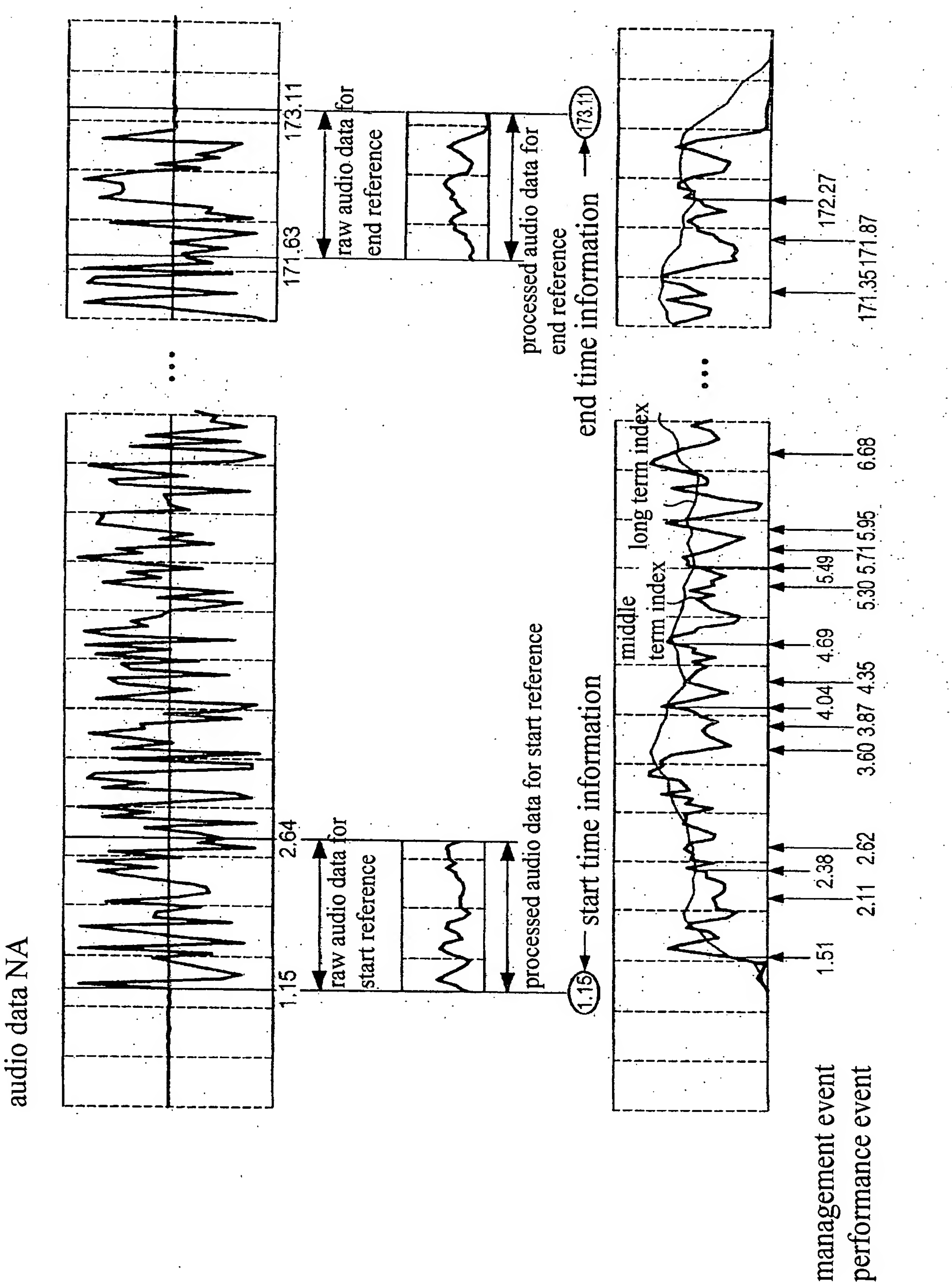
[ Fig. 12 ]


A
B
C
m
m
m
playback of performance event
0.3
0.2
0.1
0
-0.1
0.05
0.04
0.03
0.02
0.01
0
-0.01
0.02
0.01
0
-0.01
-0.02

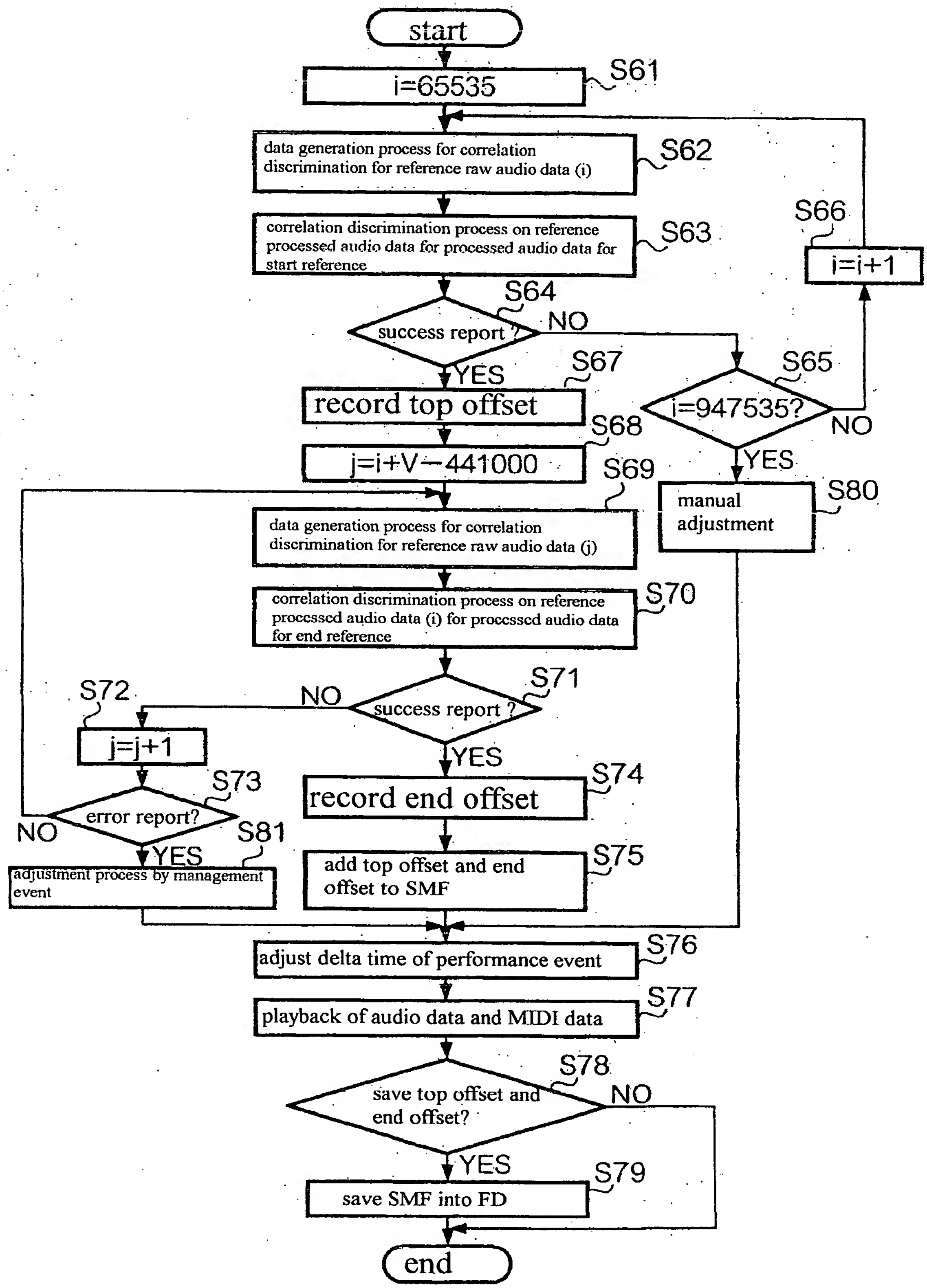
[ Fig. 13 ]


header chunk
track chunk
0.00 sec
system exclusive
processed audio data for start reference
1.15 sec
0.00 sec
system exclusive
processed audio data for end reference
173.11 sec
1.51 sec
system exclusive
management event
2.11 sec
performance event
note-on (C5)
2.38 sec
system exclusive
management event
• • •

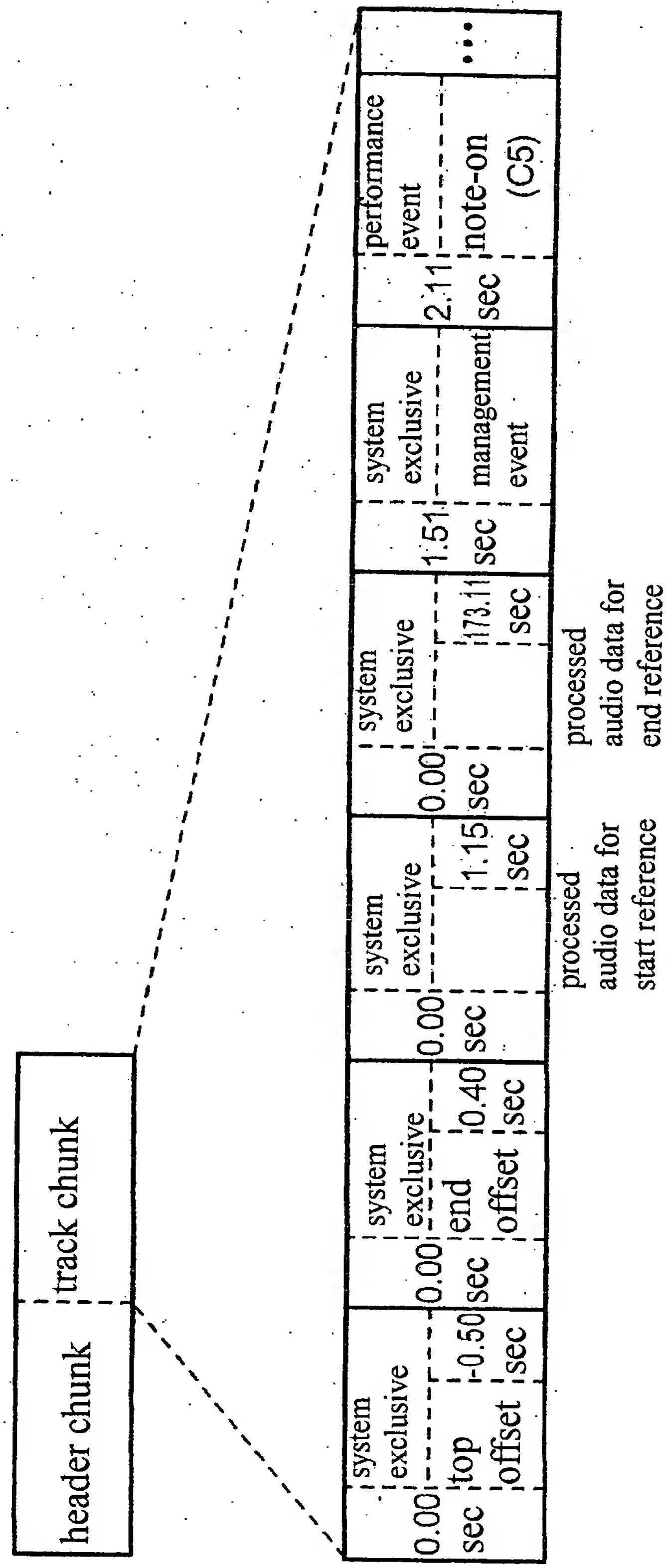
[ Fig. 14 ]


audio data NA
1.15
2.64
171.63
173.11
raw audio data for start reference
processed audio data for start reference
raw audio data for end reference
processed audio data for end reference
1.15
start time information
end time information
173.11
middle term index
long term index
1.51
2.11
2.38
2.62
3.60
3.87
4.04
4.35
4.69
5.30
5.49
5.71
5.95
6.68
171.35
171.87
172.27
management event
performance event

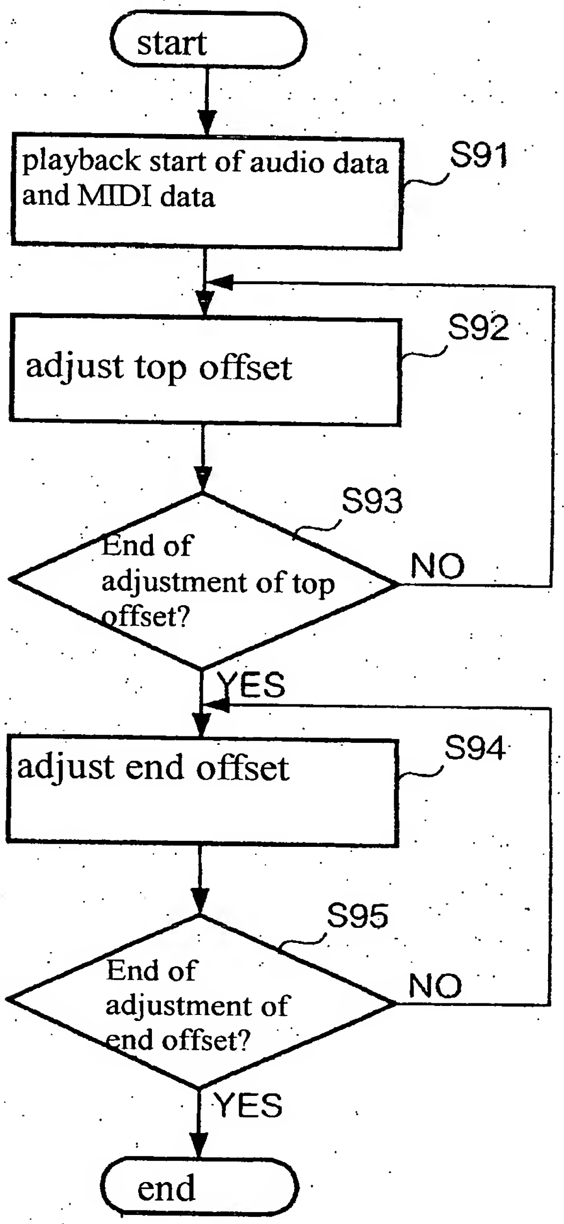
[ Fig. 15 ]


start
i=65535
S61
data generation process for correlation discrimination for reference raw audio data (i)
S62
correlation discrimination process on reference processed audio data for processed audio data for start reference
S63
S64
success report ?
NO
YES
S67
record top offset
S65
i=947535?
NO
S66
i=i+1
S68
j=i+V−441000
YES
S69
manual adjustment
S80
data generation process for correlation discrimination for reference raw audio data (j)
correlation discrimination process on reference processed audio data (i) for processed audio data for end reference
S70
S71
success report ?
NO
S72
j=j+1
S73
error report?
NO
YES
S81
adjustment process by management event
YES
S74
record end offset
add top offset and end offset to SMF
S75
adjust delta time of performance event
S76
playback of audio data and MIDI data
S77
S78
save top offset and end offset?
NO
YES
S79
save SMF into FD
end

[ Fig. 16 ]


header chunk
track chunk
0.00 sec
system exclusive
top offset
-0.50 sec
0.00 sec
system exclusive
end offset
0.40 sec
0.00 sec
system exclusive
1.15 sec
processed audio data for start reference
0.00 sec
system exclusive
173.11 sec
processed audio data for end reference
1.51 sec
system exclusive
management event
2.11 sec
performance event
note-on (C5)
• • •

[ Fig. 17 ]


start
playback start of audio data and MIDI data
S91
adjust top offset
S92
S93
End of adjustment of top offset?
NO
YES
adjust end offset
S94
S95
End of adjustment of end offset?
NO
YES
end

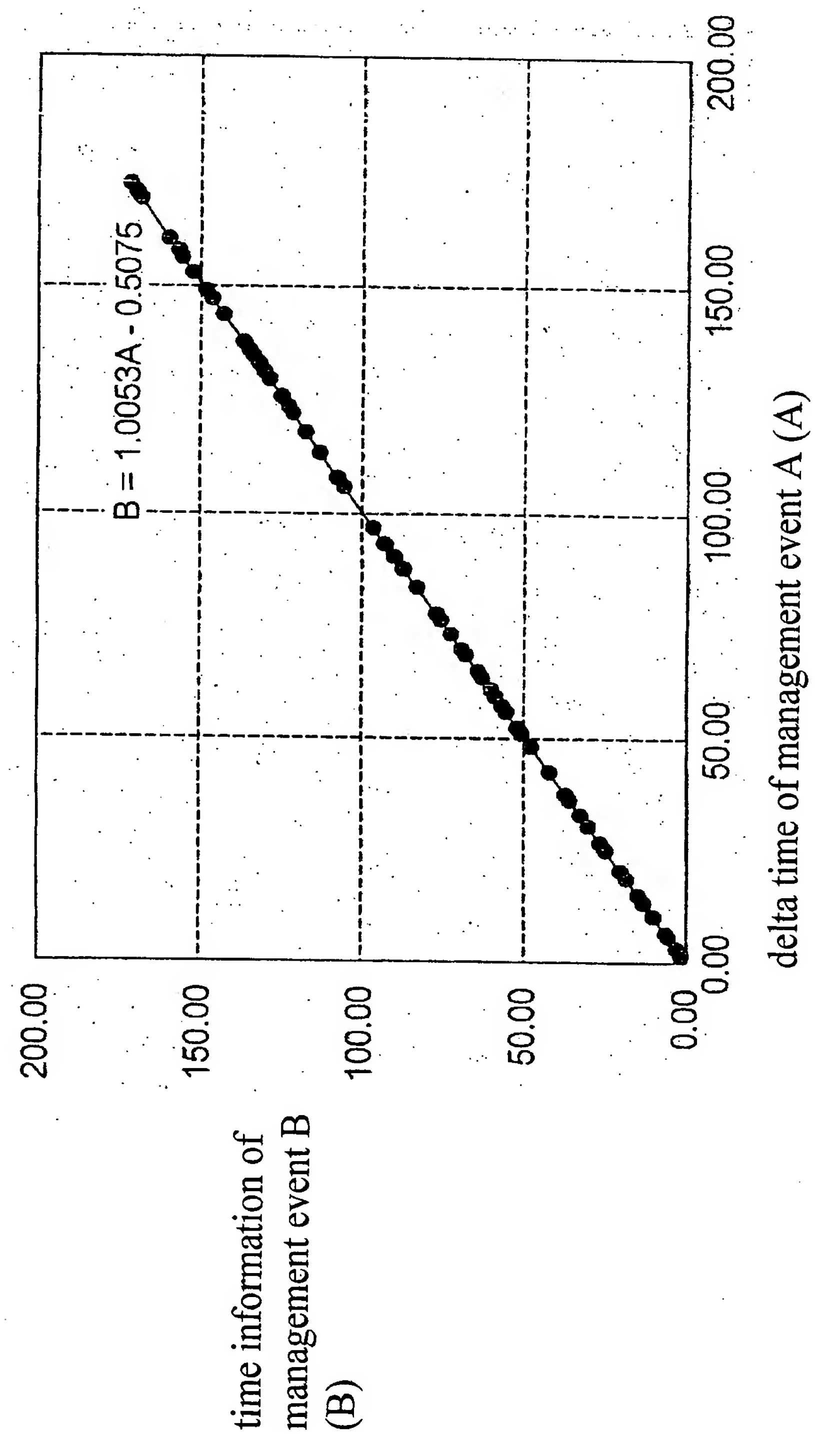
[ Fig. 18 ]

| audio data NA (SMF) | | | audio data NB | | |
|---|---|---|---|---|---|
| | | delta time (sec) | | | time information (sec) |
| 0 | start time information | 1.15 | 0 | start time information + top offset | 0.65 |
| 1 | management event A | 1.51 | 1 | management event B | 1.01 |
| 2 | management event A | 2.38 | 2 | management event B | 1.88 |
| 3 | management event A | 4.04 | 3 | management event B | 3.54 |
| 4 | management event A | 4.69 | 4 | management event B | 4.19 |
| 5 | management event A | 5.49 | 5 | management event B | 4.98 |
| 6 | management event A | 8.77 | 6 | management event B | 9.76 |
| 7 | management event A | 10.19 | 7 | management event B | 12.34 |
| 8 | management event A | 12.77 | 8 | management event B | 13.91 |
| 9 | management event A | 14.35 | 9 | management event B | 15.57 |
| 10 | management event A | 18.22 | 10 | management event B | 17.78 |
| 11 | management event A | 19.84 | 11 | management event B | 19.43 |
| 12 | management event A | 22.90 | 12 | management event B | 24.45 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 69 | management event A | 165.47 | 69 | management event B | 160.67 |
| 70 | management event A | 169.94 | 70 | management event B | 168.49 |
| 71 | management event A | 170.77 | 71 | management event B | 170.32 |
| 72 | management event A | 172.27 | 72 | management event B | 171.13 |
| | | | 73 | management event B | 171.85 |
| | | | 74 | management event B | 172.67 |

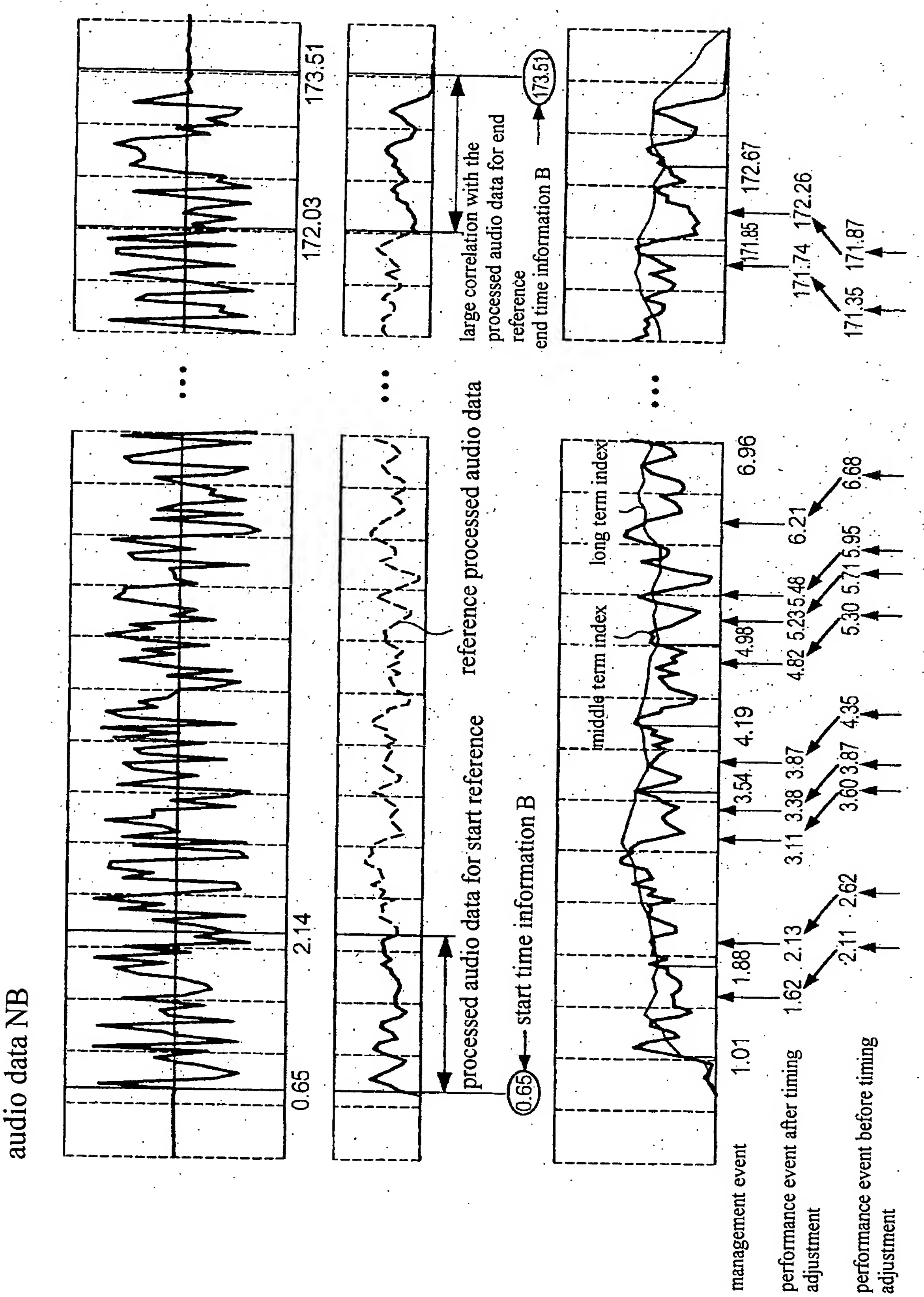
[ Fig. 19 ]

| audio data NA (SMF) | | | estimated value of time information of management event B | audio data NB | | |
|---|---|---|---|---|---|---|
| | | delta time (sec) | | | | time information (sec) |
| 0 | start time information | 1.15 | — | 0 | start time information + top offset | 0.65 |
| 1 | management event A | 1.51 | — | 1 | management event B | 1.01 |
| 2 | management event A | 2.38 | 1.88 | 2 | management event B | 1.88 |
| 3 | management event A | 4.04 | 3.54 | 3 | management event B | 3.54 |
| 4 | management event A | 4.69 | 4.19 | 4 | management event B | 4.19 |
| 5 | management event A | 5.49 | 4.99 | 5 | management event B | 4.98 |
| 6 | management event A | 8.77 | 8.25 | — | — | — |
| 7 | management event A | 10.19 | 9.67 | 6 | management event B | 9.76 |
| 8 | management event A | 12.77 | 12.35 | 7 | management event B | 12.34 |
| 9 | management event A | 14.35 | 13.93 | 8 | management event B | 13.91 |
| — | — | — | — | 9 | management event B | 15.57 |
| 10 | management event A | 18.22 | 17.79 | 10 | management event B | 17.78 |
| 11 | management event A | 19.84 | 19.41 | 11 | management event B | 19.43 |
| 12 | management event A | 22.90 | 22.51 | — | — | — |
| 13 | management event A | 24.84 | 24.46 | 12 | management event B | 24.45 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 68 | management event A | 160.34 | 160.66 | 69 | management event B | 160.67 |
| 69 | management event A | 165.47 | 165.83 | — | — | — |
| — | — | — | — | 70 | management event B | 168.49 |
| 70 | management event A | 169.94 | 170.32 | 71 | management event B | 170.32 |
| 71 | management event A | 170.77 | 171.15 | 72 | management event B | 171.13 |
| — | — | — | — | 73 | management event B | 171.85 |
| 72 | management event A | 172.27 | 172.64 | 74 | management event B | 172.67 |

[ Fig. 20 ]


B = 1.0053A - 0.5075
200.00
150.00
100.00
50.00
0.00
0.00
50.00
100.00
150.00
200.00
delta time of management event A (A)
time information of management event B (B)

[ Fig. 21 ]


audio data NB
0.65
2.14
172.03
173.51
processed audio data for start reference
reference processed audio data
large correlation with the processed audio data for end reference
0.65
start time information B
173.51
end time information B
middle term index
long term index
1.01
1.88
3.54
4.19
4.98
6.96
171.85
172.67
1.62 2.13
3.11 3.38 3.87
4.82 5.23 5.48
6.21
171.74 172.26
2.11 2.62
3.60 3.87 4.35
5.30 5.71 5.95
6.68
171.35 171.87
management event
performance event after timing adjustment
performance event before timing adjustment

[ DOCUMENT NAME ] ABSTRACT DOCUMENT

[ ABSTRACT ]

[ PROBLEM ] To provide a player, a playback method and program which permit synchronized playback of the performance data at correct timings for plural sets of audio data of the same musical tune having different start timings and end timings.

[ SOLVING MEANS ] A controller 6 plays back performance data stored in an SMF that is read out from an FD drive 2 at the same timing of the playback of the audio data transmitted from a music CD drive 1. Time information representing a start timing and an end timing of a musical tune in audio data recorded in the SMF. When the playback of the audio data and the playback of the performance data are shifted, the user instructs adjustment of a start timing and an end timing of the performance data to the controller 6 by using the manipulation display 5. The controller 6 follows the adjustment instruction given by the user, modifies the time information representing the start timing and the end timing of the musical tune stored in the SMF, and adjusts the playback timing of the performance data by using the modified time information.

[ SELECTED FIGURE ] Fig. 1